

Universitá degli Studi di Salerno

Dipartimento di Informatica

Tesi di Laurea di I livello in

Informatica

Skill Alexa per l'interrogazione di Knowledge Graph

Relatori

Prof. Vittorio Scarano

Dott.ssa Maria Angela Pellegrino

Candidato

Mario Santoro

0512104850

Anno Accademico 2018-2019

I computer sono incredibilmente veloci, accurati e stupidi.
Gli uomini sono incredibilmente lenti, inaccurati e intelligenti.
L'insieme dei due costituisce una forza incalcolabile.
(Albert Einstein)

Abstract

L'avvento dell'era digitale ha portato ad un'esplosione di informazioni sottoforma di notizie, articoli, social media e così via. Ogni individuo dotato di un dispositivo elettronico produce inconsapevolmente, ogni giorno, migliaia di dati.

É necessario modellare i dati in modo che possano essere interrogati facilmente. Un approccio possibile è modellarli attraverso un Knowledge Graph (con l'acronimo KG). Sebbene non vi sia una definizione chiara di KG, un'interpretazione comune è che esso rappresenta una raccolta di descrizioni interconnesse di entità: oggetti del mondo reale, eventi, situazioni o concetti astratti. Spesso contengono grandi volumi di informazioni fattuali con semantica meno formale. In alcuni contesti, il termine Knowledge Graph viene utilizzato per riferirsi a qualsiasi Knowledge base rappresentata come grafo. Esso costituisce dunque un ambiente volto a facilitare la raccolta, l'organizzazione e la distribuzione della conoscenza. Un'ulteriore classificazione distingue KG proprietari e quelli gestiti da community, possono essere specializzate o general purpose.

Ma per interrogare i KG sono necessari linguaggi come SPARQL che si rivelano essere non alla portata di tutti. SPARQL si è affermato tra i principali strumenti di interrogazione nonostante la sua complessità. Nasce così la necessità di rendere accessibili questi dati anche per persone senza competenze tecniche in query language, in generale, e SPARQL, in particolare.

Quindi come fare per creare un servizio per l'interrogazione di Knowledge Graph attraverso interfacce facilmente usabili e che non richiedano competenze tecniche? Tra i diversi approcci che semplificano l'interazione utente ci sono gli assistenti vocali, molto popolari negli ultimi anni, che rappresentano un modo facile e intuitivo per l'interazione in linguaggio naturale e per l'erogazione di servizi di un'applicazione.

Secondo una previsione degli analisti britannici di Juniper Research, l'uso degli assistenti vocali è destinato a triplicare nei prossimi anni. L'azienda stima che ci saranno 8 miliardi di assistenti vocali digitali in uso entro il 2023, rispetto ai 2,5 miliardi del 2018. É quindi evidente che l'utilizzo della tecnologia di riconoscimento vocale è la prossima grande frontiera. Amazon Alexa è la tecnologia oggi in testa alle classifiche con una quota di mercato del 73%.

Per questi motivi l'assistente vocale scelto è Alexa, mentre la Knowledge Graph scelta per l'interrogazione è Wikidata, definita come una knowledge base online collaborativa, sostenuta e ospitata dalla Wikimedia Foundation, scelta per l'efficacia delle API fornite.

Quindi un utente effettua delle domande ad un assistente vocale in linguaggio naturale, la skill riconosce le parole chiave, viene generata la query SPARQL sul Knowledge Graph e vengono recuperate le informazioni richieste.

Che tipo di domande può effettuare l'utente? Le domande sono modellate basandoci sulla cosiddetta regola delle 5 W, cioè la regola principale dello stile giornalistico anglosassone e delle regole di buona formazione del discorso, Who, What, When, Where, Why. Al fine di una valutazione corretta della skill si è scelto di effettuare testing su un dataset ufficiale di domande risposte su Wikidata.

L'obiettivo di tesi è dunque: semplificare l'utilizzo di KG senza richiedere nessuna competenza tecnica, mediante la creazione di una skill Alexa per permettere l'interrogazione (in linguaggio naturale) di un Knowledge Graph, cioè Wikidata.

Indice

1 Introduzione **7**

2 Background **9**

2.1 Alexa Skill 9

2.1.1 Modello di interazione 9

2.1.2 Back-end 11

2.2 Knowledge Graph 11

2.3 SPARQL 12

2.4 Wikibase SDK 13

3 Personal assistant based on Wikidata **15**

3.1 Pattern 16

3.2 Workflow 16

3.3 Use Case 18

3.3.1 getResult 18

3.3.2 getDescription 19

3.3.3 getLocation 21

3.3.4 getSuperlative 22

3.3.5 getSuperlativeFilter 23

3.3.6 getQuantitive 23

3.3.7 getResultReverse 24

3.3.8 getNumericFilter 25

3.4 Problemi e Risoluzioni 27

3.4.1 Gestione di più risultati 27

3.4.2 Mancanza di Wikidata property 27
3.4.3 Gestione dei sinonimi 28
3.4.4 Gestione ambiguità 29
3.4.5 Problema del timeout 31
3.4.6 Caratteri speciali 31
3.5 Echo Show e Firestick 31
3.5.1 getImg 32
3.6 Modello di interazione 33
3.7 Back-end 36
3.8 Pubblicazione 39
3.8.1 Problema slot 39

4 Testing **42**
4.1 Valutazione 42
4.2 Risultati 54

5 Conclusioni e sviluppi futuri **57**
5.1 Sviluppi Futuri 57
5.1.1 DBpedia 58

Elenco delle figure

2.1 Architettura skill Alexa 10
2.2 Esempio Knowledge Graph 12

3.1 Workflow ideale 16
3.2 Workflow reale 17
3.3 Use case - getResult 18
3.4 Use case - getResult Temporali 19
3.5 Use case - getDescription 20
3.6 Use case - getLocation 21
3.7 Use case - getSuperlative 22
3.8 Use case - getSuperlativeFilter 23
3.9 Use case - getQuantity 24
3.10 Use case - getResultReverse 25
3.11 Use case - getNumericFilter 26
3.12 Use case - getResult Gestione di più risultati 27
3.13 Mancanza di Wikidata property nell'entità 27
3.14 Problema dei sinonimi 28
3.15 Risoluzione problema ambiguità 30
3.16 Timeout nella Lambda function 31
3.17 Use case - getImg 33

CAPITOLO 1

Introduzione

Secondo alcuni dati estrapolati da un più recente articolo su Forbes, ogni giorno, generiamo 2,5 quintilioni di gigabyte di dati. Le stime dicono che nel 2020 ogni persona sulla terra genererà 1,7 megabyte di dati ogni secondo. Una delle opzioni per modellare questa vasta quantità di dati sono i Knowledge Graph (KG).

Nell'ultimo decennio, infatti, centinaia di dataset sono stati pubblicati utilizzando lo Standard web semantico che coprono una varietà di domini [3]. Questi dataset sono descritti utilizzando il modello di dati RDF, che si basa su grafi. Questi modelli descrivono i concetti e le relazioni su di essi attraverso l'introduzione di triple soggetto-predicato-oggetto, consentendo la costruzione di query basate su triple. L'idea di utilizzare grafi per modellare e gestire diversi dati su larga scala è andata oltre la comunità del Semantic Web, poichè sono diventati sempre più popolari, segnati dai recenti annunci di vari Knowledge Graph [5]. Infatti su Lod Cloud, un Knowledge Graph di Knowledge Graph, c'è stato un incremento notevole dei KG presenti, partendo da 12 KG nel 2007 ai 1239 KG presenti nel 2019. Alcuni di questi Knowledge Graph sono proprietari, gestiti internamente da società ad esempio Google, Microsoft, Apple, ecc.; mentre altri sono aperti al pubblico via Web, gestiti da comunità internazionali dedicate, come DBpedia, Wikidata, ecc.

Tra gli standard del Web Semantico si è affermato SPARQL per l'interrogazione di KG. SPARQL è un querying language che permette il recupero delle informazioni da KG e la modellazione dei risultati in tabelle (o sotto-grafi). Il problema principale di SPARQL e di altri querying language su KG è la loro complessità

Tuttavia, interrogare i grafi usando questi linguaggi può essere impegnativo. Primo, gli utenti devono avere una conoscenza tecnica di tali linguaggi di query e la semantica dei loro operatori. In secondo luogo, i grafi vengono spesso utilizzati per rappresentare dati diversi che potrebbero non corrispondere a un particolare schema specifico del dominio, nel senso che gli utenti potrebbero non essere in grado di concettualizzare facilmente i dati che stanno interrogando. Per sfruttare appieno i KG nasce l'esigenza di fornire strumenti che mascherino le difficoltà dei linguaggi di interrogazione, quali SPARQL. Tra i diversi approcci abbiamo valutato l'utilizzo di smart speaker data la loro popolarità e semplicità di interazione. Dunque in questo lavoro ho verificato se (e fino a che punto) gli smart speaker possano semplificare l'interrogazione di KG.

A tal fine, è stata progettata e implementata una skill Alexa per estrarre delle informazioni da una Knowledge Base, Wikidata, innescata da una domanda in linguaggio naturale dall'utente all'assistente vocale.

La tesi è organizzata come segue: nel secondo capitolo, viene illustrato il background. Nel terzo capitolo viene illustrata la progettazione della skill e la sua implementazione. Nel quarto capitolo viene descritta la valutazione riportando il dataset testato e i risultati ottenuti. Infine, nel quinto capitolo vengono presentate le conclusioni e gli sviluppi futuri.

CAPITOLO 2

Background

In questo capitolo sarà descritto il dominio applicativo del lavoro di tesi, relativo alla skill Alexa. In particolare, verranno illustrate in dettaglio tutte le caratteristiche che la compongono. Inoltre vedremo cosa sono e come si interrogano i Knowledge Graph.

2.1 Alexa Skill

Le skill sono le "abilità", ovvero le applicazioni che aggiungono capacità ad Alexa. Ogni cosa che Alexa può fare viene implementata in una Skill. Essa dispone di alcune skill predefinite e disponibili in tutte le lingue supportate, come ad esempio l'orologio, le liste, i timer e i promemoria. Tutte le altre, invece, sono sviluppate da terze parti e raccolte in un catalogo del tutto simile ad un application store. Entrando nello specifico, una skill si compone essenzialmente di due parti:

- Modello di interazione;
- Back-end.

Nella figura 2.1 viene mostrata l'architettura della Skill, con il dispositivo smart speaker che accede al modello di interazione, quest'ultimo tramite endpoint si collega alla parte back-end. Vediamo ora in dettaglio le due parti della skill.

2.1.1 Modello di interazione

Il modello di interazione consiste in un file json che modella le caratteristiche della skill:

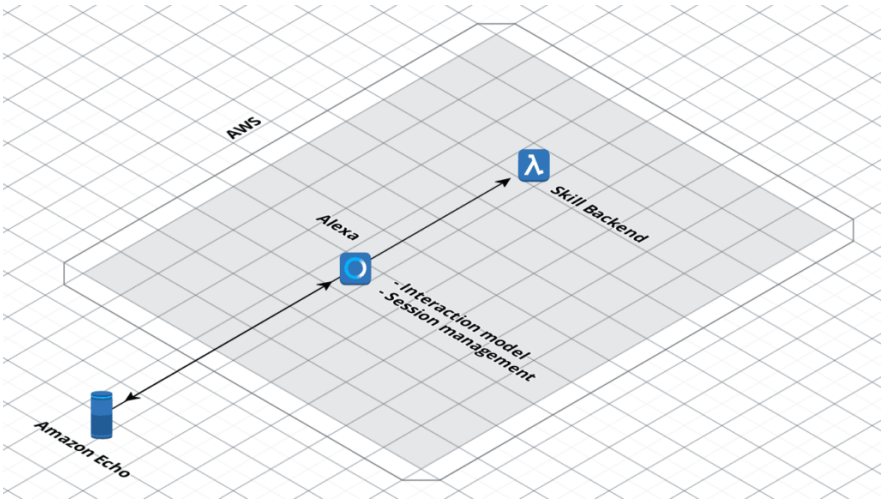


Figura 2.1: Architettura skill Alexa

- **Definizione dell'invocazione** [1]: L'invocazione è usata per avviare una skill. Questa deve essere molto chiara e significativa, qualcosa che l'utente possa ricordare in quanto dovrà usare il nome della skill per avviarlo. Idealmente il nome di una skill dovrebbe essere composto da almeno due parole e corrispondere all'invocazione completa, per esempio "Alexa, apri Mio Servizio". Qui "Mio Servizio" rappresenta il nome della skill, mentre alcune parole diverse possono essere usate al posto di "apri", come "chiedi" (ampiamente usato), "inizia", "fai", "lancia", "carica", ecc;

- **Intent** [1]: Questa è una terminologia tecnica, un intent rappresenta ogni funzione che la nostra skill svolgerà. Gli utenti possono invocare intent diversi per eseguire funzioni diverse. Per esempio se una skill accende/spegne la luce può avere un intent per accendere/spegnere ed un intent per aggiungere un nuovo utente;

- **Utterance** [1]: sono formulazioni alternative delle frasi che potremo utilizzare per richiamare lo stesso intent. Mentre si costruisce una skill Alexa è possibile scrivere manualmente gli enunciati delle frasi che l'utente potrà dire per richiamare un particolare intent, o una particolare funzionalità all'interno della nostra skill. Sarà inoltre necessario scriverne quante più possibile per ottimizzare al massimo la comprensione precisa degli intent degli utenti. Ad esempio se la skill in questione fornisse gli orari della prima colazione di un hotel i diversi enunciati potrebbero essere: "Qual è l'orario di colazione?" "A che ora inizia il servizio di prima colazione?" "Quando è la prima colazione?" "A che ora termina il servizio

di prima colazione?" "Vi prego di farmi sapere gli orari di colazione", ecc;

- **Slot** [1]: sono i parametri della richiesta fornita dall'utente. Ad esempio, riprendendo il caso di una skill per conoscere gli orari dei pasti di un hotel, la utterance potrebbe essere: "a che ora inizia pasto", dove pasto è uno slot. Quando l'utente chiederà "a che ora inizia _la colazione_?" lo slot pasto viene istanziato da _la colazione_. Alexa Skills Kit supporta diversi tipi di slot che definiscono il modo in cui i dati nello slot vengono riconosciuti e gestiti. I tipi forniti rientrano nelle seguenti categorie generali: numeri, date e orari, frasi, elenchi di articoli, ecc. Gli slot vengono inseriti nelle utterance, quando quest'ultima viene invocata viene fatto il matching con i valori possibili di quello slot.

Tutte queste caratteristiche sono strettamente collegate: l'utente effettua una richiesta, se la richiesta fa il match con una utterance, viene identificato l'intent richiesto per poi allocare gli slot con parametri effettivi, utilizzabili nel back-end.

2.1.2 Back-end

Per il back-end Amazon offre una delle piattaforme leader nel cloud computing Amazon Web Services (AWS). Uno dei servizi AWS è consigliato da Amazon per connettere la propria Skill al cloud: si tratta di AWS Lambda. Questo è un servizio che si prende cura dell'esecuzione del codice senza doversi preoccupare di gestire un server. Inoltre, Lambda gestisce automaticamente le risorse necessarie a eseguire il codice, rendendo semplice scalare il servizio erogato. Alexa Skills Kit è un kit di sviluppo software (SDK) che consente a uno sviluppatore di sviluppare competenze, chiamate anche applicazioni di conversazione, sull'assistente di intelligenza artificiale Amazon Alexa[2]. Una skill Alexa tramite un endpoint si collega ad una Amazon Lambda Services dove viene implementata la logica dell'applicazione.

La logica back-end può essere implementata in qualsiasi linguaggio di programmazione. É possibile inserire librerie esterne. L'interfaccia pone vincoli sulla dimensione del file direttamente modificabili dalla piattaforma, ma permette il caricamente di file zip di dimensione arbitraria.

2.2 Knowledge Graph

Una knowledge Graph (KG) è una tecnologia usata per memorizzare informazioni complesse in modo strutturato o non strutturato. Nickel et al. hanno definito un KG come

segue: "[...] un grafo strutturato knowledge base (KB) che memorizza informazioni fattuali in forma di relazione tra entità"[5]. Un KG è costituito da una base di conoscenza che rappresenta fatti sul mondo e un motore di inferenza che può ragionare su tali fatti e utilizzare regole e altre forme di logica per dedurre nuovi fatti o evidenziare incoerenze [7]. Un Knowledge Graph è una specie di rete semantica, un artefatto il cui ambito, caratteristiche, persino usi, rimangono aperti e sono in procinto di essere definiti. Tra i KG più noti annovero Wikidata, una knowledge base online collaborativa, sostenuta e ospitata dalla Wikimedia Foundation, con lo scopo di fornire un database comune di supporto agli altri progetti di Wikimedia, consultabile e modificabile sia da utenti umani che da bot. È affine a Wikimedia Commons nel modello di fruizione, poiché fornisce dati a tutti i progetti e consente loro un accesso centralizzato, esattamente come fa Commons con i file multimediali [10]. Il linguaggio RDF descrive i concetti e le relazioni su di essi attraverso l'introduzione di triple (soggetto-predicato-oggetto), e consente la costruzione di query basate su triple patterns, congiunzioni logiche, disgiunzioni logiche, e pattern opzionali. Un KG è un grafo, con archi etichettati che sono le proprietà, i nodi sono le entità. Nella figura 2.2 viene mostrato come esempio il grafo con le informazioni inerenti al compositore Johann Sebastian Bach, viene identificato tramite URI e attraverso gli archi (proprietà) si riesce a risalire al nome, ai figli e altre informazioni a esso collegate.

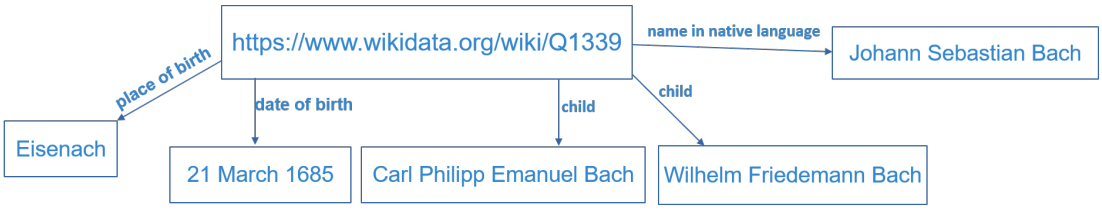


Figura 2.2: Esempio Knowledge Graph

2.3 SPARQL

Per poter interrogare Wikidata si utilizza **SPARQL**, un come query language. Esso è uno degli elementi chiave delle tecnologie legate al paradigma noto come web semantico, e consente di estrarre informazioni dalle basi di conoscenza distribuite sul web. Le query in SPARQL sono basate su triple pattern dove i parametri da recuperare sono sostituite da variabili, introdotte da '?'. [9]. Supponiamo di voler elencare tutti i figli del compositore barocco Johann Sebastian Bach la query in SPARQL è la seguente:

```
SELECT ?child WHERE {
```

```
  # ?child  father   Bach
  ?child wdt:P22 wd:Q1339.
}
```

Su Wikidata, gli oggetti e le proprietà non sono identificati da nomi leggibili dall'uomo come "padre" (proprietà) o "Bach" (oggetto). (Per una buona ragione: "Johann Sebastian Bach" è anche il nome di un pittore tedesco, e "Bach" potrebbe anche riferirsi al cognome, al comune francese, al cratere di Mercurio, ecc.). Per evitare ambiguità, i valori nei KG sono identificati mediante URI. Le URI di Wikidata sono di due tipologie, e si riferiscono a oggetti e predicati, assegnati a identificatori, cioè Q seguito da un numero per gli oggetti (o entità), e P seguito da un numero per i predicati (o proprietà). **WDQS** è il servizio di query di Wikidata si occupa dell'inserimento di una query SPARQL, la esegue sul set di dati di Wikidata e ne mostra il risultati [6]. Per ogni URI deve essere definito il prefisso associato. Per le triple WDQS, gli oggetti devono essere preceduti dal prefisso wd: e i predicati con wdt:.

2.4 Wikibase SDK

Per interrogare Wikidata attraverso il backend di Alexa abbiamo utilizzato **Wikibase SDK**, è una suite di strumenti Javascript per interrogare un'istanza di Wikibase e ottenere i risultati. Questo pacchetto è stato sviluppato principalmente come wikidata-sdk ma poi è stato generalizzato per supportare qualsiasi istanza di Wikibase: wikidata.org tra gli altri. Esso utilizza JavaScript ES6, che significa NodeJS dalla versione 6.4.0 o successive o browser Web non troppo obsoleti. Per installare in locale i moduli necessari si utilizza il comando "npm install wikibase-sdk" sulla Command-line interface. Per poter utilizzare i moduli installati facciamo l'import con il seguente codice in Node.js:

```
const wbk = require('wikibase-sdk')({
  instance: 'https://www.wikidata.org',
  sparqlEndpoint: 'https://query.wikidata.org/sparql'
})
```

Le API di Wikibase sono un insieme di funzioni per eseguire query di lettura su un'istanza di Wikibase, e sono suddivise in 4 categorie prinicpali [4]

- Cerca entità
- Ottieni entità;
  - Per id;
  - Per ID e revisione;

  - Dai titoli di Wikipedia;
  - Con altri titoli di progetti Wikimedia.
- Semplifica i dati delle entità;
- Ottieni revisioni.

In particolare 'Cerca entità' permette di recuperare l'identificativo di un item dando come parametro il nome dell'item in linguaggio naturale.

```
const search = 'Ingmar Bergman'
const language = 'fr' /* will default to 'en' */
const limit = 10 /* defaults to 20 */
const format = 'json' /* defaults to json */

const url = wbk.searchEntities(search, language, limit, format)
```

Infine esistono funzioni aggiuntive per le istanze di Wikibase che hanno un servizio di query SPARQL (come Wikidata Query per wikidata.org). Per ottenere risultati JSON da una query SPARQL è possibile effettuare una richiesta HTTP a https://query.wikidata.org/sparql?query=SPARQL&format=json , che con wikidata-sdk può essere eseguita in questo modo:

```
  //Inizializzazione di WBK con un sparqlEndpoint
const wbk = require('wikibase-sdk')({
      instance: 'https://my-wikibase-instan.se',
      sparqlEndpoint: 'https://query.my-wikibase-instan.se/sparql'
})
  const sparql = 'SELECT * WHERE { ?s ?p ? o } LIMIT 10'
  const url = wbk.sparqlQuery(sparql)
   //  richiede l'URL generato per effettuare una richiesta HTTP
       utilizzando una libreria a scelta
 request({ method: 'GET', url })
```

CAPITOLO 3

Personal assistant based on Wikidata

Per realizzare l'obiettivo stabilito, ovvero creare un servizio per facilitare l'interrogazione di Knowledge base attraverso interfacce facilmente usabili e che non richiedano competenze tecniche, abbiamo scelto di interagire con gli utenti tramite smart spaker di Amazon, Alexa. In particolare come Knowledge Base si è scelto Wikidata per l'usabilità e chiarezza delle API fornite. In generale l'obiettivo è quindi creare una skill Alexa per interrogare Wikidata. La skill da implementare è una skill per ottenere risposte a delle domande in linguaggio naturale poste dall'utente, ma che tipo di domande? Basandoci sulla cosiddetta regola delle 5 W, cioè la regola principale dello stile giornalistico anglosassone e delle regole di buona formazione del discorso, le cinque W stanno per:

- Who? [≪Chi?≫]
- What? [≪Che cosa?≫]
- When? [≪Quando?≫]
- Where? [≪Dove?≫]
- Why? [≪Perché?≫]

Sono considerati i punti irrinunciabili che devono essere presenti nella prima frase di ogni articolo, come risposta alle probabili domande del lettore che si accinge a leggere il pezzo. Su questa idea vengono modellate le domande che la skill può rispondere [8].

3.1 Pattern

In particolare le query su cui ci andremo a soffermare sono query SELECT e i pattern inizialmente identificati per l'implementazione sono i seguenti:

- Who is ⟨entity⟩? → Who is Turing?
- What is the ⟨property⟩ of ⟨entity⟩? → What is the mass of sun?
- When is the ⟨property⟩ of ⟨entity⟩? → When is the date of birth of Chopin?
- Where is ⟨entity⟩? → Where is Mona Lisa?
- What is the ⟨entity⟩ with ⟨superlative⟩ ⟨property⟩? → What is river with maximum length?
- How many ⟨entity⟩ has ⟨property⟩ like ⟨entityTwo⟩? → How many film has directed like tim barton ?

Per i superlativi si è ricorso ad una semplificazione, ovvero l'utilizzo delle parole "maximum/highest" per identificare un superlativo di maggioranza e le parole "minimum/lowest" per identificare un superlativo di minoranza.

3.2 Workflow

Utilizzando le API Wikidata vorremo avere un workflow come nella figura 3.1.

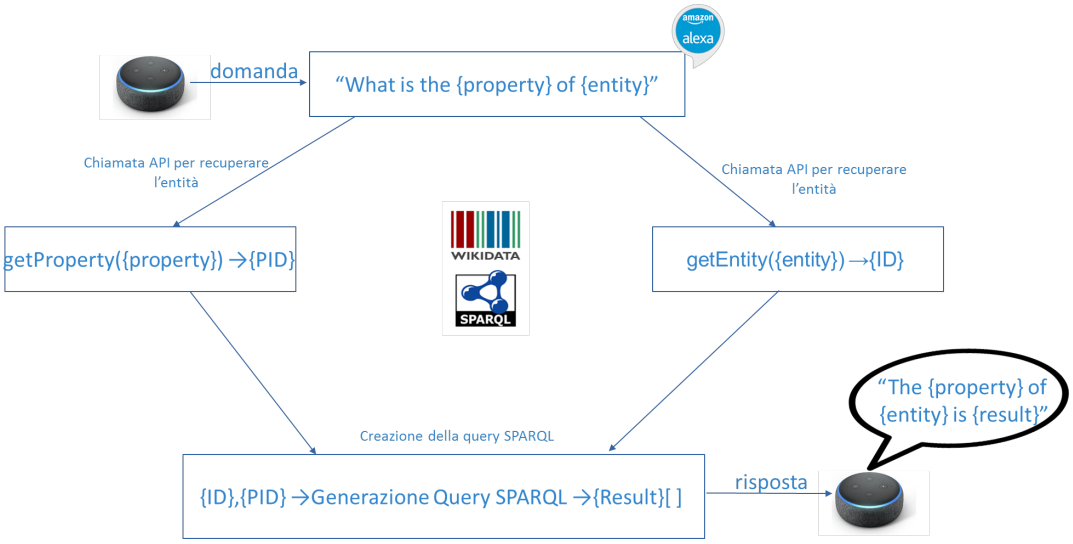


Figura 3.1: Workflow ideale

Cioè recuperare tramite API sia la proprietà che l'entità dalla frase in input e in seguito generare una query SPARQL che utilizzi gli identificativi ottenuti dalle chiamate API. Però spesso la realtà differisce dall'ideale. Infatti nella figura 3.2 viene mostrato il workflow reale. Non essendoci una funzione fornita dalle API in grado di recuperare

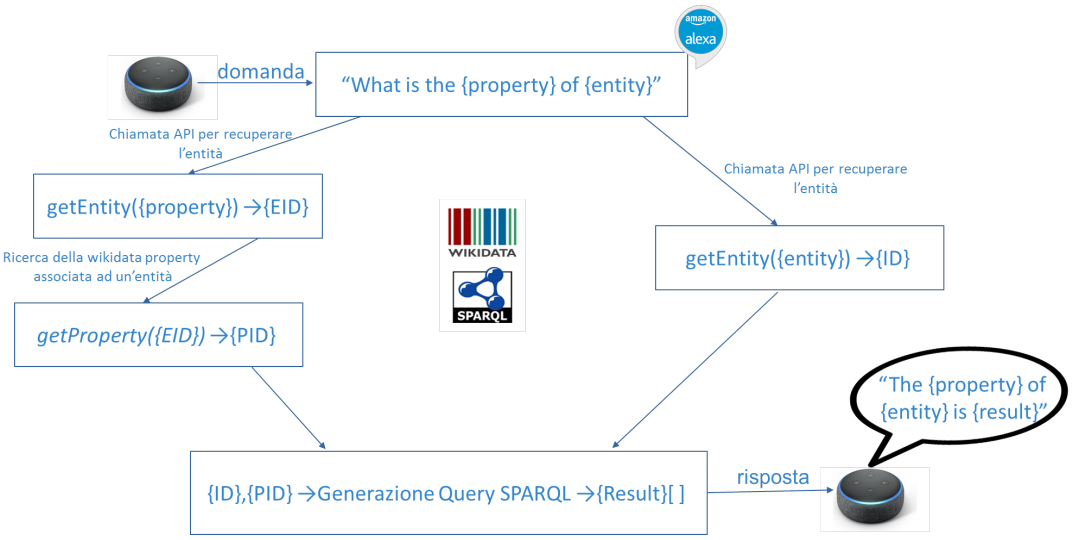


Figura 3.2: Workflow reale

l'identificativo per le proprietà la soluzione trovata si avvale del fatto che ogni informazione è memorizzata in Wikidata come entità, quindi anche le proprietà. Queste entità particolari hanno associate una proprietà chiamata "Wikidata property" contenente l'identificativo della proprietà che ci occorre per la creazione della query. Fornendo un esempio più chiaro, in Wikidata esiste una proprietà "length" ed esiste anche un'entità "length" in esso contenuta come proprietà "Wikidata property" con l'identificativo della proprietà "length". Focalizzandoci sulla funzione getProperty, essa effettua una query SPARQL:

```
SELECT ?id WHERE {
    wd:"+EID+" wdt:P1687 ?id.
}
```

La proprietà P1687 è la proprietà "Wikidata property" che associa all'entità "EID" l'URL della proprietà effettiva. Dopodiché i valori ottenuti, entità e proprietà vengono passati a una funzione che genera la query SPARQL per ottenere la risposta alla domanda posta dall'utente.

3.3 Use Case

In questa sezione vediamo i vari casi d'uso dei pattern implementati e le SPARQL query per ottenere il risultato. Gli esempi forniti nelle immagini seguenti sono fatti da test reali sulla skill nella Alexa Developer Console.

3.3.1 getResult

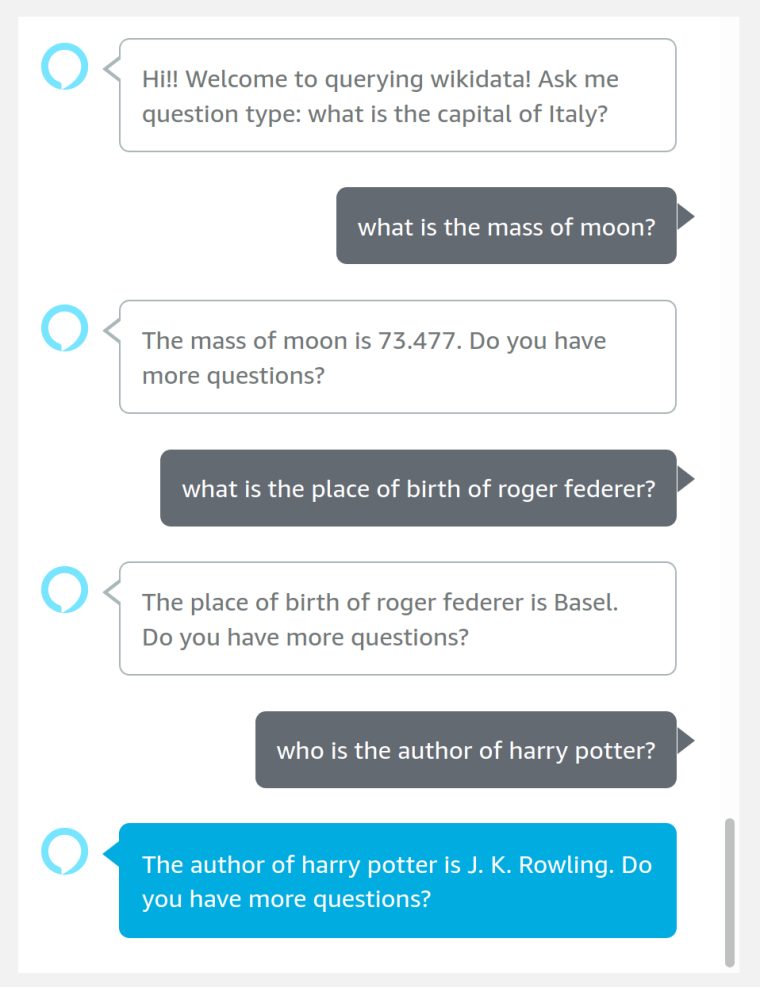


Figura 3.3: Use case - getResult

Nella figura 3.3 vengono mostrati alcuni esempi su domande che possono essere risposte con questo pattern.

Oltre a rispondere domande con "what" o "who" è stato accorpato anche il pattern definito inizialmente "temporale" cioè le domande con "when" o in cui viene chiesto un dato temporale come output mostrato nella figura 3.4. In quanto anche queste ultime possono essere risposte con la medesima struttura query, nel seguente modo:



Figura 3.4: Use case - getResult Temporali

```
SELECT ?result ?resultLabel WHERE {
   wd:"+entity+"  wdt:"+property+" ?result;
   SERVICE wikibase:label {
        bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
        }
 }
```

3.3.2 getDescription

Il pattern per le descrizioni non utilizza query SPARQL per ottenere la risposta alle domande, bensì la funzione fornita dalle API per recuperare le entità restituisce l'oggetto JSON con già la descrizione dell'entità stessa.

```
{"searchinfo":
    {   "search":"Paris"},
        "search":[
        {"repository":"",
         "id":"Q90",
         "concepturi":"http://www.wikidata.org/entity/Q90",
         "title":"Q90",
         "url":"//www.wikidata.org/wiki/Q90",
```

```
        "label":"Paris",
        "description":"capital and largest city of France",
        "match":{"type":"label","language":"en","text":"Paris"}
      },
                        ...
  ],"search-continue":10,"success":1}
```


Quindi recuperiamo la descrizione dall'oggetto JSON nel modo seguente:

```
var res = request('GET', entity);
var b = JSON.parse(res.getBody());
var ids=b.search[0].description;
return ids;
```

Nella figura 3.5 vengono mostrati alcuni esempi.

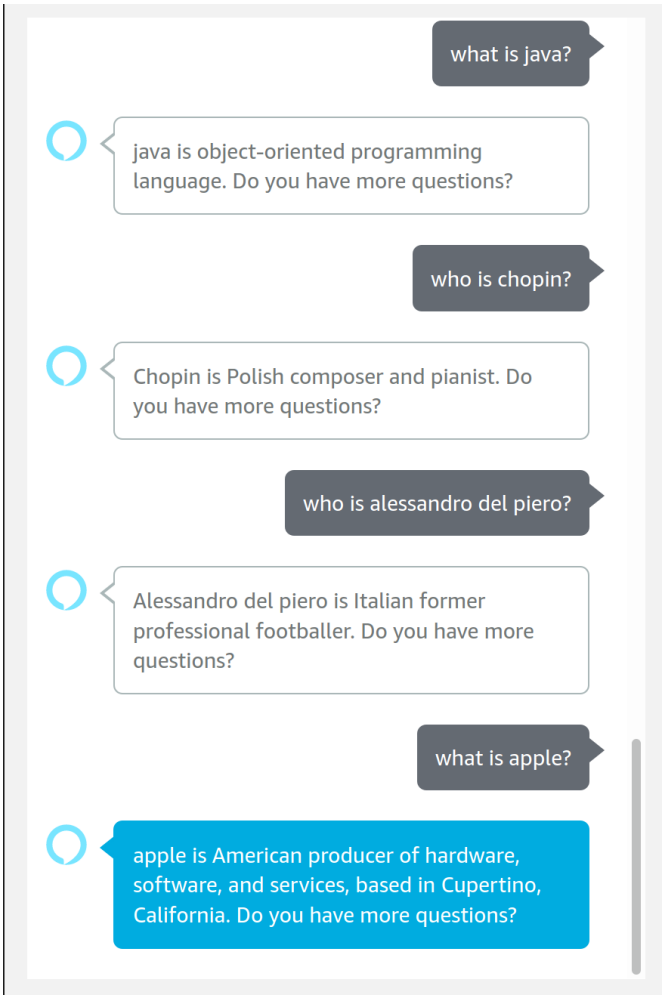


Figura 3.5: Use case - getDescription

3.3.3 getLocation

Nel pattern "getLocation" appena recuperata l'entità tramite API viene generata la query SPARQL seguente per ottenere una risposta:

```
SELECT ?result ?resultLabel ?loc ?locLabel WHERE {
  VALUES ?loc {wdt:P276 wdt:P6375 wdt:P17}
  wd:"+entity+" ?loc ?result;
    SERVICE wikibase:label {
      bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
    }
} ORDER BY DESC(?result)
```

In questa query l'entità è associata a tre tipi di proprietà di locazione di Wikidata, in questo modo il risultato ottenuto può essere ordinato in maniera decrescente per avere un luogo dal più specifico (là dove è presente l'informazione) al più generico. Nella figura 3.6 vengono mostrati alcuni esempi.

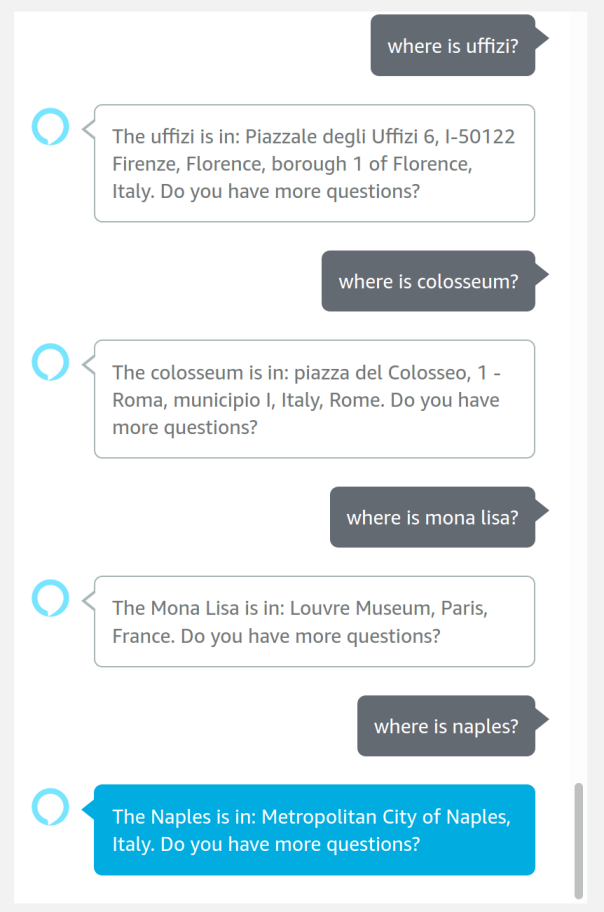


Figura 3.6: Use case - getLocation

3.3.4 getSuperlative

In "getSuperlative" recuperiamo dagli slot tre parole chiave, la proprietà e l'entità di cui recuperare gli identificativi e il superlative, maximum/highest oppure minimum/lowest, quest'ultimo è fondamentale per stabilire l'ORDER BY della query SPARQL:

```
SELECT ?result ?resultLabel  ?res WHERE{
    VALUES ?a {wd:"+entity+"}
    VALUES ?p {wdt:"+property+"}
    ?result wdt:P31 ?a.
    ?result ?p ?res.
    SERVICE wikibase:label{
        bd:serviceParam wikibase:language "en"
    }
}ORDER BY "+ord+"(?res) LIMIT 1
```

La proprietà P31 occorre per recuperare tutte le istanze di quell'entità, dopodichè sul risultato viene associata la property. Siccome il risultato che ci occorre sarà sempre il primo, grazie all'ORDER BY, ne andiamo a dare un LIMIT 1 perchè vogliamo solo quello che corrisponde al valore minimo o massimo. Esempi nella figura 3.7.

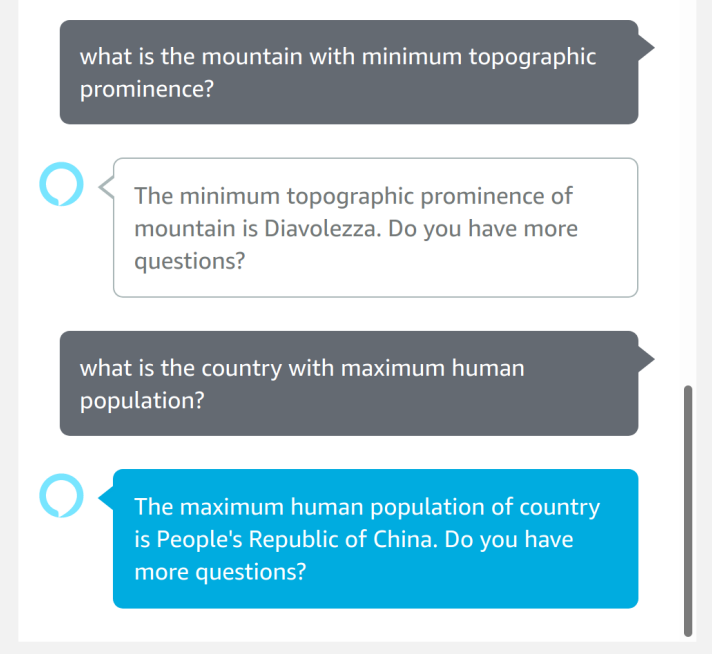


Figura 3.7: Use case - getSuperlative

3.3.5 getSuperlativeFilter

Questo pattern è una specializzazione del pattern getSuperlative spiegato nel sottoparagrafo 3.3.4. La differenza è che viene aggiunta una entità di locazione per limitare la ricerca in un'area, come mostrato dagli esempi in figura 3.8.

La query SPARQL invece viene modellata nel modo seguente:

```
SELECT ?result ?resultLabel  ?res WHERE{
    VALUES ?a {wd:"+ids+"}
    VALUES ?p {wdt:"+p+"}
    ?result wdt:P31 ?a.
    ?result wdt:P17 wd:"+filter+".
    ?result ?p ?res.
    SERVICE wikibase:label {
        bd:serviceParam wikibase:language \"en\"
    }
}ORDER BY "+ord+"(?res) LIMIT 1
```

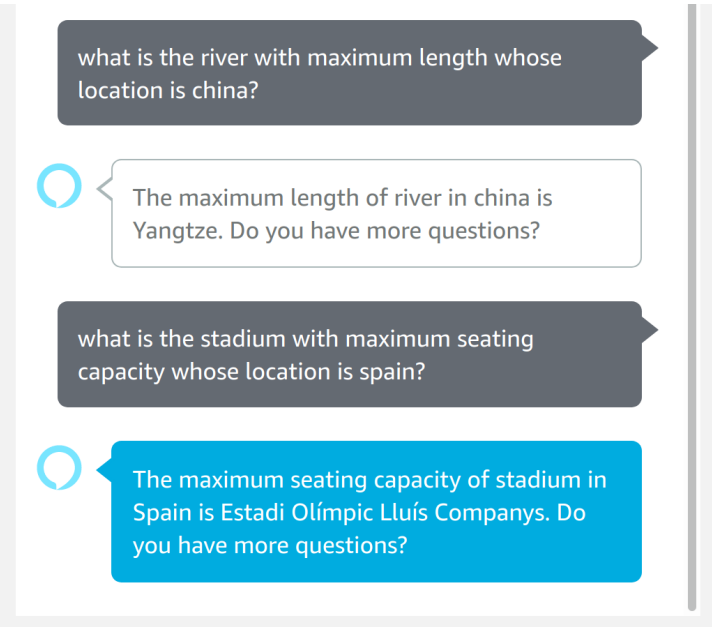


Figura 3.8: Use case - getSuperlativeFilter

3.3.6 getQuantitive

Nel pattern Quantitativo viene specificata sia la quantità dei risultati trovati, sia l'elenco dei primi 5 risultati trovati (in caso di minor risultati vengono elencati tutti). Vengono mostrati nella figura 3.9 alcuni esempi. In questo pattern vengono recuperate tre parole chiave: entity e filter, che fanno riferimento allo stesso slot ovvero entity, e poi property

che fa riferimento allo slot property. I tre identificativi vengono recuperati con le apposite funzioni, dopodichè la query SPARQL viene generata nel seguente modo:

```
SELECT ?result ?resultLabel WHERE {
    ?result  wdt:P31 wd:"+entity+";
    wdt:"+property+" wd:"+filter+".
    SERVICE wikibase:label {
        bd:serviceParam wikibase:language \"[AUTO_LANGUAGE],en\".
    }
}
```

La proprietà P31 recupera tutte le istanze di entity, dopodichè ne associa la property e il filter passati in input.

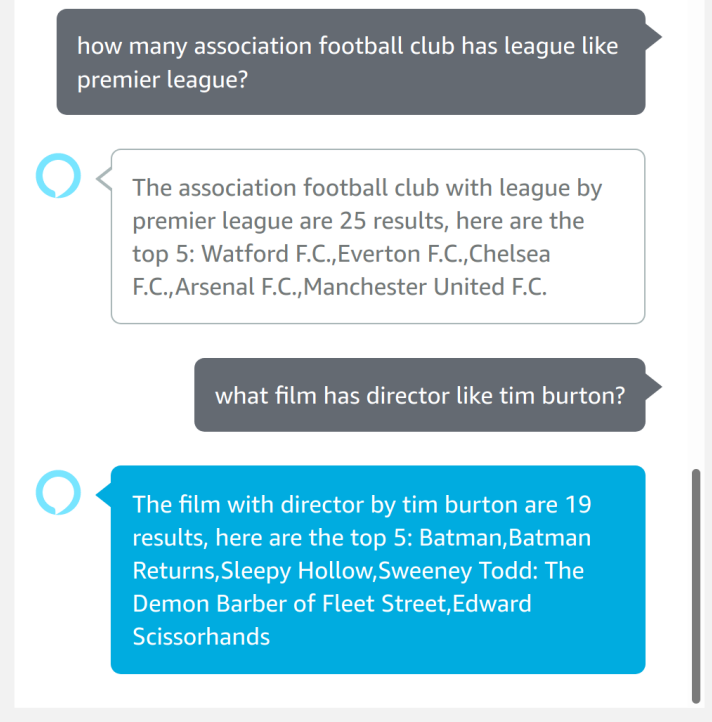


Figura 3.9: Use case - getQuantity

3.3.7 getResultReverse

Le domande risposte dal pattern getResult nel sottoparagrafo 3.3.1 risponde a domande modellate con soggetto-predicato-oggetto ma in alcuni casi le domanda potrebbero essere formulante in maniera inversa, cioè partendo dall'oggetto. Per questi casi speciali è stato creato questo pattern con la query SPARQL nel seguente modo:

```
SELECT ?result ?resultLabel WHERE {
    ?result  wdt:"+p+" wd:"+ids+";
```

```
    SERVICE wikibase:label {
        bd:serviceParam wikibase:language \"[AUTO_LANGUAGE],en\".
    }
}
```

Alcuni esempi sono forniti nella figura 3.10.

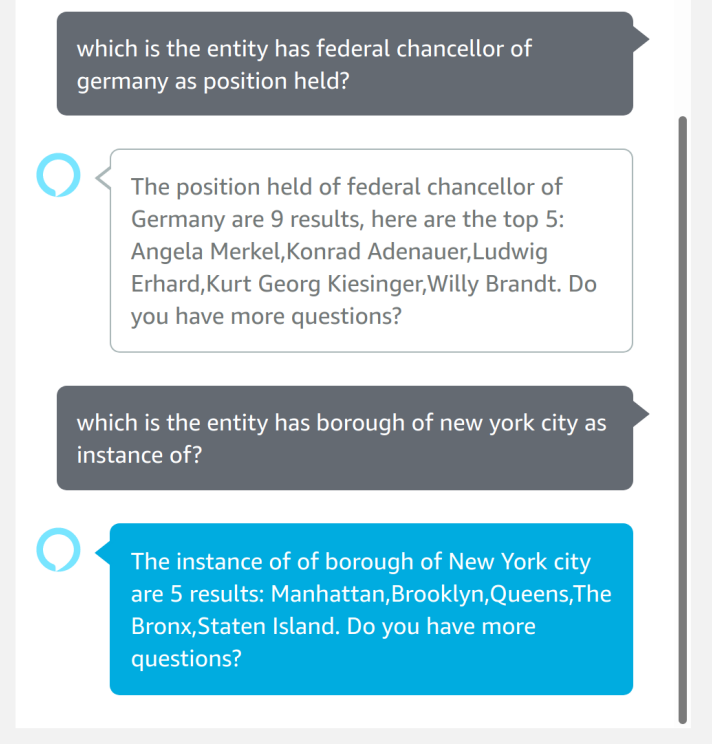


Figura 3.10: Use case - getResultReverse

3.3.8 getNumericFilter

In questo caso le domande che possono essere risposte con questo pattern sono quelle con un filtro numerico. I parametri di input sono 5, vengono individuate prima una serie di entità (entity) associate ad una proprietà (property) e in seguito di queste si vuole quelle che hanno una proprietà (propertyTwo) che sia minore/maggiore/uguale (symbol) a un valore numerico (value) che viene recuperato tramite la slot fornita da Amazon, cioè AMAZON.NUMBER. Formando una domanda generica come: "Which person has the same ⟨property⟩ as ⟨entity⟩ and her ⟨propertyTwo⟩ is ⟨symbol⟩ to ⟨value⟩?". La query SPARQL per queste domande possono essere di due tipi, ci sono quelle che devono essere raggruppate per anno, se vogliamo imporre un vincolo su una data, e altre invece che il vincolo numerico può essere generico e quindi un semplice FILTER. Per rispondere a entrambe le tipologie di domande con un solo intent si è scelto di effettuare

un'interrogazione a cascata, dove nel caso di fallimento (nessun risultato trovato) della prima struttura, viene lanciata la seconda. Le due strutture sono le seguenti:

```
SELECT DISTINCT ?result ?resultLabel WHERE {
    ?result wdt:"+p+" wd:"+ids[i]+".
    ?result wdt:"+p2+" ?value .
    FILTER(?value "+simbolo+value+" )
    SERVICE wikibase:label { bd:serviceParam wikibase:language \"en\" }}
```

```
SELECT DISTINCT ?result ?resultLabel WHERE {
    ?result wdt:"+p+" wd:"+ids[i]+".
    ?result wdt:"+p2+" ?value .
    FILTER(year(?value) "+simbolo+value+" )
    SERVICE wikibase:label {
        bd:serviceParam wikibase:language \"en\"
    }
}
```

Alcuni esempi di utilizzo nella figura 3.11.

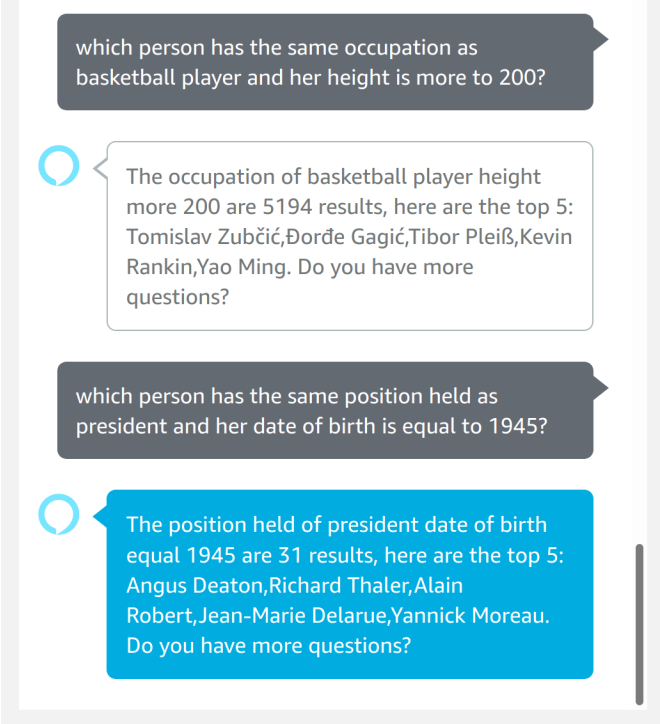


Figura 3.11: Use case - getNumericFilter

3.4 Problemi e Risoluzioni

3.4.1 Gestione di più risultati

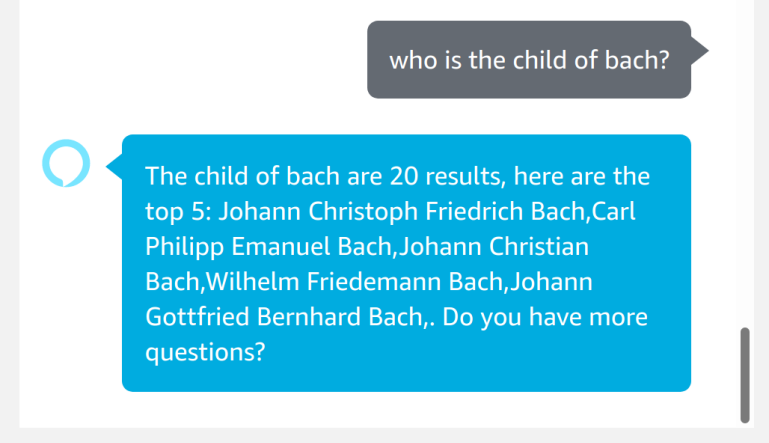


Figura 3.12: Use case - getResult Gestione di più risultati

Per tutti i pattern in caso di una lista di risultati le informazioni sono state gestite in modo da elencare tutti i risultati se sono minore o uguale a 5, altrimenti se superiori Alexa dirà il numero di risultati totali ma elencherà solo i primi 5 trovati (esempio nella figura 3.12).

3.4.2 Mancanza di Wikidata property

Alcune domande, nonostante il pattern implementato per risponderle fosse corretto, non ottenevano una risposta. Il problema era la mancanza della proprietà "Wikidata property" associata ad alcune entità, ad esempio wife e husband. La soluzione è stata quella di inserire manualmente questa proprietà nelle entità trovate su Wikidata come nell'esempio in figura sottostante per l'entità husband che deve collegarsi alla proprierà "spouse".

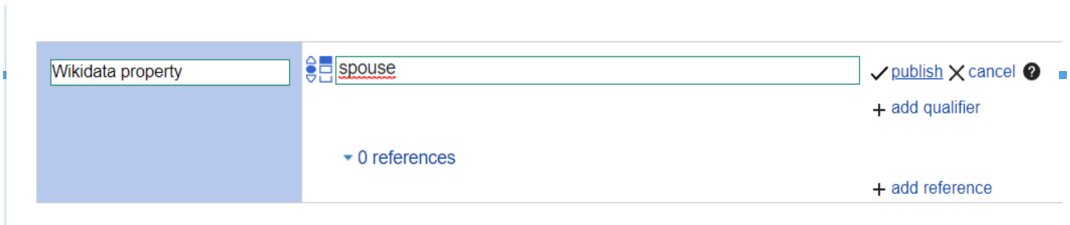
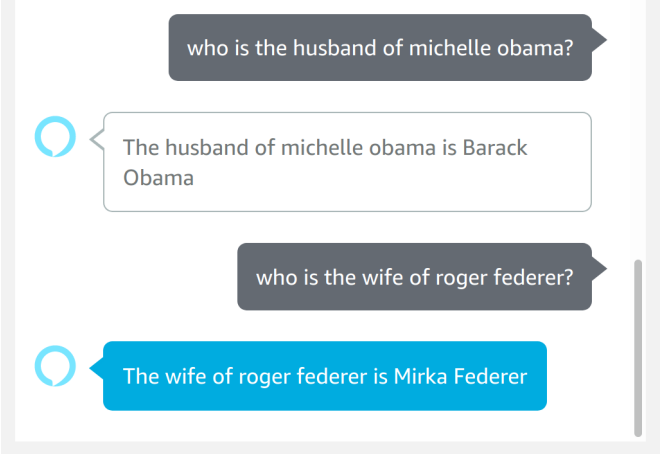


Figura 3.13: Mancanza di Wikidata property nell'entità

Infatti aggiungendo quelle proprietà alle due entità, le domande prima non risposte da Alexa vengono trovate come mostrato nella figura seguente.



3.4.3 Gestione dei sinonimi

Ulteriore problematica riscontrata è la non gestione dei sinonimi nella funzione getEntity, ad esempio nella figura 3.14 per ottenere il luogo di nascita di Chopin sono stati usati 3 sinonimi e la skill è riuscita a rispondere 2 volte su 3.

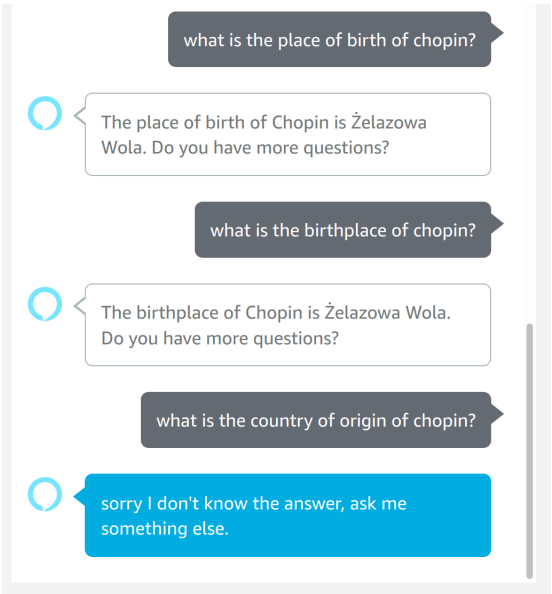
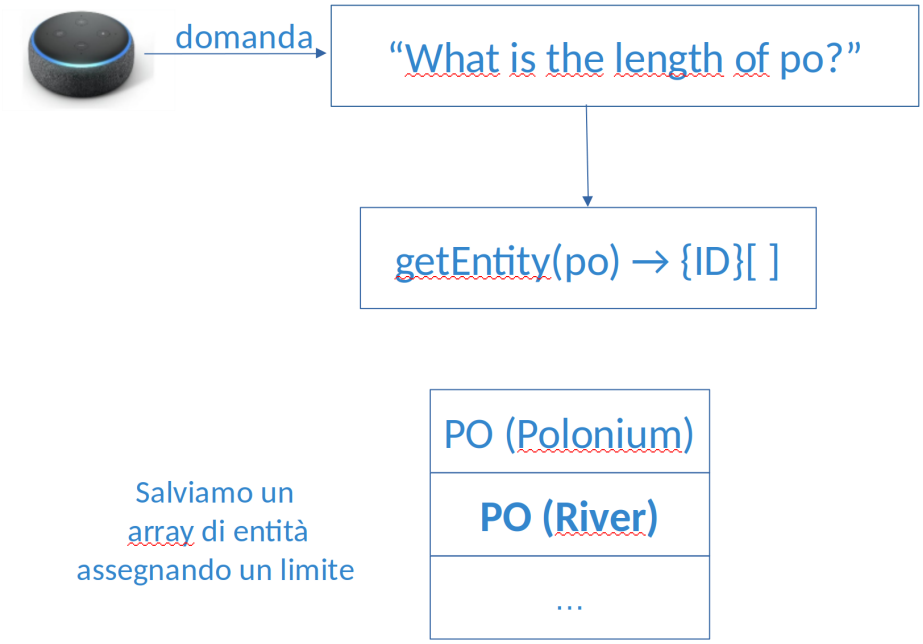


Figura 3.14: Problema dei sinonimi

3.4.4 Gestione ambiguità

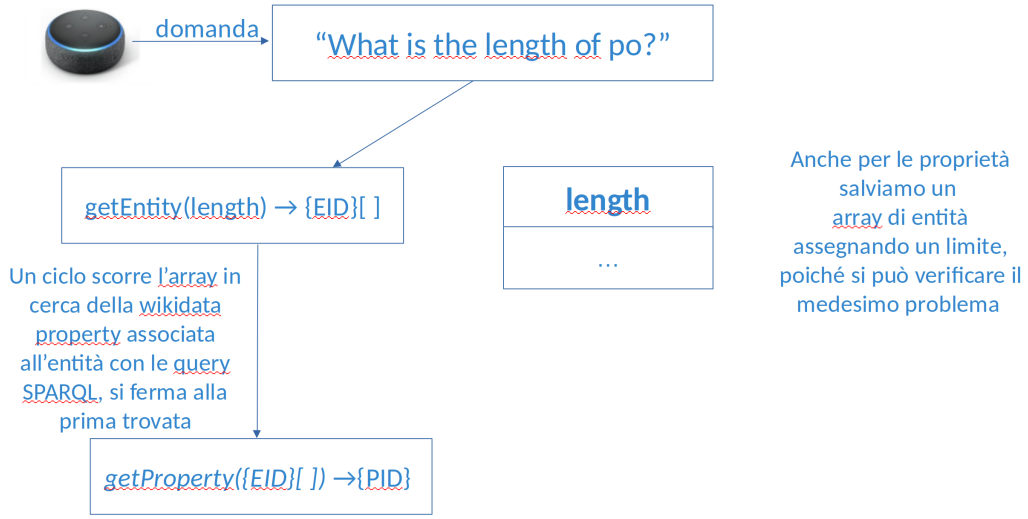
Facendo alcune prove sulla skill si sono palesate diverse domande non risposte per un motivo comune. Prendendo come esempio la domanda "What is the length of po?", quando vengono recuperati gli identificativi tramite API, l'entità po che viene recuperata è quella relativa a PO di Polonium e non Po il fiume.

Tutti i risultati dopo la chiamata API su una certa entità sono in una lista in un oggetto JSON, l'idea per risolvere il problema quindi è, anzichè prendere il primo valore restituito dall'oggetto JSON come fatto fino a quel momento, salvare in un array la lista dei risultati contenuti nell'oggetto (stabilendo un limite, nella skill è stato messo un limite di 10 entità). In questo modo:



Come mostrato nella figura nell'array abbiamo sia PO di Polonium sia Po il fiume, insieme ad altri Po intesi come altri concetti trovati in Wikidata.

Il passo successivo e disambiguare anche le proprietà, che possono riscontrare il medesimo problema. Come mostrato nel paragrafo 3.2 nella figura 3.2 le proprietà vengono intese in un primo momento come entità, quindi ugualmente vengono salvati in un array i risultati della chiamata API sulla proprietà. Dopodichè nella funzione, creata in back-end con la query SPARQL per recuperare le Wikidata property, viene modificata aggiungendo un ciclo per provare a recuperare la wikidata property per l'array di entità memorizzato, come mostrato in figura:



Appena la query, utilizzando l’i-esima entità dell’array, trova una proprietà associata esce dal ciclo e restituisce in output la proprietà trovata, questo perchè dopo aver effettuato un po’ di testing è risultato che il primo valore con una wikidata property associata era quello giusto.

Infine la proprietà e l’array di entità trovate vengono date in input alla funzione con la query dinamica SPARQL per ottenere i risultati. Questa funzione viene modificata con un ciclo che per ogni entità genera una query SPARQL associandone la proprietà, come un matching per trovare una corrispondenza tra i due valori, appena viene riscontrato un risultato il ciclo termina restituendo le informazioni trovate (figura 3.15).

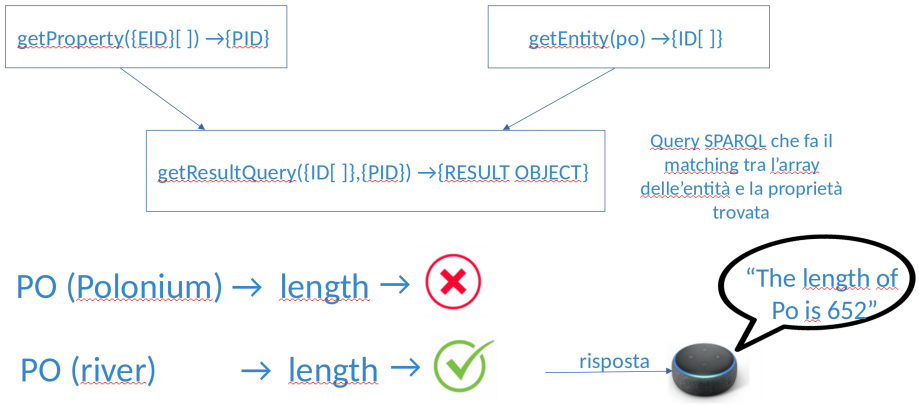


Figura 3.15: Risoluzione problema ambiguità

3.4.5 Problema del timeout

Alcune domande possono risultare onerose in termini di tempo, e ciò comportava ad un'improvvisa terminazione della skill, in quanto veniva superato il timeout della Lambda function nel backend. Su AWS nella lambda function è possibile modificare facilmente il timeout inserendo un limite che soddisfi le proprie richieste (figura 3.16).

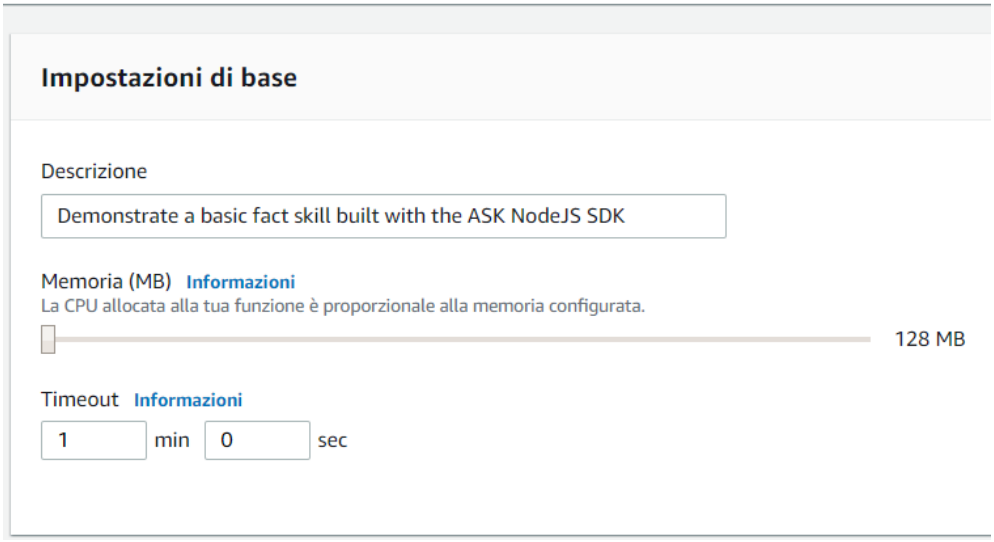


Figura 3.16: Timeout nella Lambda function

3.4.6 Caratteri speciali

Alcune risposte a domande chieste dall'utente possono contenere caratteri speciali (&, /, _, -) ciò comporta ad un improvvisa terminazione della skill in quanto il dispositivo vocale non sa come pronunciare questi caratteri. Per evitare questi tipi di problemi è stata creata una funzione che, dato un risultato, verifica se in esso vi è un'occorrenza di un carattere speciale, in questo caso il valore viene sostituito (con un 'and' se il carattere è '&', con uno spazio vuoto per gli altri), altrimenti se non sono presenti questi caratteri il risultato non viene alterato.

3.5 Echo Show e Firestick

Echo Dot è ottimo per le risposte vocali ma se volessimo visualizzare delle immagini? Amazon ha due dispositivi per visualizzare immagini:

- **Amazon Echo Show** uno smart display da 10 pollici, dotato di connettività, che integra l'assistente vocale Amazon Alexa. Controllabile con la voce, è in grado di interagire con gli utenti, eseguire diversi comandi vocali oltre che mostrare i vari contenuti direttamente sullo schermo touch;
- **Amazon Firestick** un dongle HDMI, cioè una sorta di chiavetta che si collega alla porta HDMI del televisore (oltre che alla presa della corrente) e permette di accedere a una vasta gamma di contenuti online comprese le skill Alexa.

Nasce l'idea di implementare un pattern che renda possibile visualizzare immagini prese da Wikidata attraverso domande poste al dispositivo Alexa.

3.5.1 getImg

Questo nuovo pattern recupera l'entità con la stessa chiamata API di quelli precedentemente visti. Però occorre fare delle distinzioni sul codice in back-end.

Appena recuperato il valore nell'handler viene lanciata la funzione per controllare se il dispositivo è supportato per visualizzare immagini:

```
  if (supportsDisplay(handlerInput)) {
      const display_type = "BodyTemplate7"
      const image_url = res;
      response = getDisplay(handlerInput.responseBuilder, attributes
          , image_url, display_type)

        return response
      .speak(`this is a '+entity+". Do you have more questions?")
       .reprompt(`more question?')
      .getResponse();
}else{
  response = handlerInput.responseBuilder
        return response
  .speak(`you don\'t have a device that displays images. Do you have
       more questions?')
  .reprompt(`more question?')
  .getResponse();
```

Nel caso non è un dispositivo di questo tipo, viene lanciato un messaggio di avviso all'utente, altrimenti viene visualizzata l'immagine richiesta. La query SPARQL che permette di recuperare le immagini è la seguente:

```
SELECT ?img ?imgLabel WHERE{
    wd:"+ids+" wdt:P18 ?img;
    SERVICE wikibase:label{
```

```
        bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
    }
}
```

La proprietà P18 è immagine, e si riferisce all'entità passata come parametro alla funzione apposita.

Il risultato finale è un intent che permette la visualizzazione di immagini sia su dispositivi e Echo Show che Firestick, nella figura 3.17 è possibile vedere un esempio nell'Alexa Developer Console.

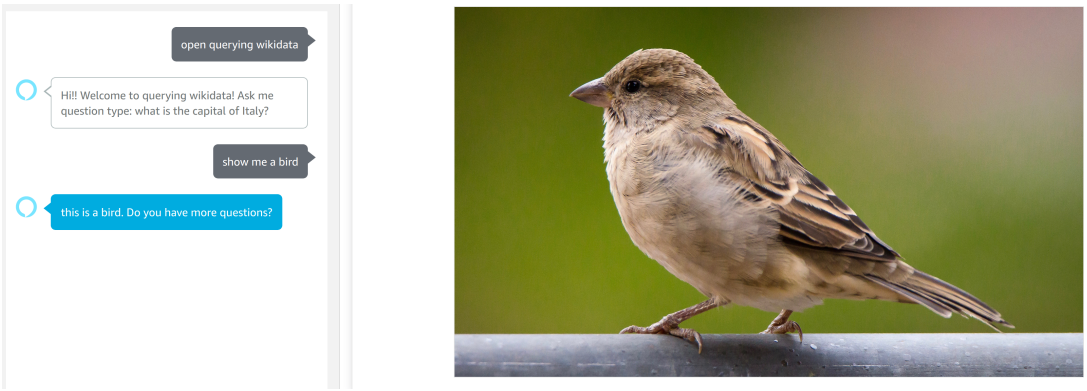


Figura 3.17: Use case - getImg

3.6 Modello di interazione

Per coprire più domande possibili si è scelto di fornire un intent diverso per ogni struttura query identificata.

- Intent: getResult
  - Utterances
    - when is the ⟨property⟩ of ⟨entity⟩;
    - who is the ⟨property⟩ of ⟨entity⟩;
    - what is the ⟨property⟩ of ⟨entity⟩.
  - Slots
    - entity;
    - property.
- Intent: getDescription

  - Utterances
    - who is ⟨entity⟩;
    - what is ⟨entity⟩.
  - Slots
    - entity.

- Intent: getLocation
  - Utterances
    - where is ⟨entity⟩;
    - where is the ⟨entity⟩.
  - Slots
    - entity.

- Intent: getSuperlative
  - Utterances
    - what is the ⟨entity⟩ with ⟨superlative⟩ ⟨property⟩.
  - Slots
    - entity;
    - property;
    - superlative (maximum/highest o minimum/lowest).

- Intent: getResultFilter (nome dato all'intent getQuantity)
  - Utterances
    - how many ⟨entity⟩ has ⟨property⟩ like ⟨filter⟩.
    - what ⟨entity⟩ has ⟨property⟩ like ⟨filter⟩.
  - Slots
    - entity (filter usa slot entity);
    - property.

- Intent: getImg
  - Utterances
    - show me ⟨entity⟩.

        * show me a ⟨entity⟩.
        * show me an ⟨entity⟩.
        * show me the ⟨entity⟩.
    - Slots
        * entity

- Intent: getResultReverse
    - Utterances
        * Which is the entity has ⟨entity⟩ as ⟨property⟩.
    - Slots
        * entity
        * property

- Intent: getSuperlativeFilter
    - Utterances
        * what is the ⟨entity⟩ with ⟨superlative⟩ ⟨property⟩ whose location is ⟨filter⟩
    - Slots
        * entity (anche filter usa lo slot entity)
        * property
        * superlative

- Intent: getNumericFilter
    - Utterances
        * which person has the same ⟨property⟩ as ⟨entity⟩ and her ⟨propertyTwo⟩ is ⟨symbol⟩ to ⟨value⟩?
    - Slots
        * entity
        * property (usato anche da propertyTwo)
        * symbol
        * AMAZON.NUMBER (usato da value)

3.7 Back-end

Nella Lambda Function il linguaggio di programmazione utilizzato è Node.js, vediamo come esempio l'handler di risposta all'intent *"getResult"*:

```
const getResult_Handler =  {
    canHandle(handlerInput) {
        const request = handlerInput.requestEnvelope.request;
        return request.type === 'IntentRequest' && request.intent.
            name === 'getResult' ;
    },
    handle(handlerInput) {
        const request = handlerInput.requestEnvelope.request;
        const responseBuilder = handlerInput.responseBuilder;
        let sessionAttributes = handlerInput.attributesManager.
            getSessionAttributes();

        let say = 'The ';

        let slotStatus = '';
        let resolvedSlot;
        var entity='';
        var property='';
        let slotValues = getSlotValues(request.intent.slots);

        /* getSlotValues restituisce .heardAs, .resolved e .
            isValidated per ogni slot, in base ai codici di stato
            degli slot di richiesta ER_SUCCESS_MATCH,
            ER_SUCCESS_NO_MATCH o slot di richiesta semplice
            tradizionale senza risoluzioni*/

                //   SLOT: property
        if (slotValues.property.heardAs) {
            //controlli sullo slot
                .
                .
                .
             //   SLOT: entity
        if (slotValues.entity.heardAs) {
            //controlli sullo slot
                .
                .
                .
        }
```

```
            say += slotStatus;

            var p=getProperty(property); //funzione per recuperare
                identificativo della property

            var ide= getEntity(entity); //funzione per recuperare
                identificativo del entity

            const res =  getResultQuery(ide,p); /* funzione che prende
                in input gli identificativi di entity e propriety e crea
                dinamicamente la query SPARQL per ottenere la risposta
                alla domanda in input*/

            say += res;

            return responseBuilder
                .speak(say+". Do you have more questions?")
                .reprompt('more question?')
                .getResponse();
        },
    };
```

Infine esaminiamo la funzione *"getResultQuery"* chiamata dall'handler *"getResult_Handler"*:

```
function getResultQuery(ids,p){
    var c='';
    var sparql="
        SELECT ?citta ?cittaLabel WHERE {
            wd:"+ids+"  wdt:"+p+" ?citta;
            SERVICE wikibase:label {
                bd:serviceParam wikibase:language \"en\".
            }
        }";
    const url = wbk.sparqlQuery(sparql);

    var request = require('sync-request');
    var res = request('GET', url, {
      headers: {
        'user-agent': 'example-user-agent',
      },
    });

    var b= JSON.parse(res.getBody());
```

```
    var count = Object.keys(b.results.bindings).length;

    if(count>5){
        c+=‘are ’+count+‘ results, here are the top 5: ’;
        for(var f=0; f<5; f++){
            c+=b.results.bindings[f].cittaLabel.value;
            if(f!=count-1){
                c+=‘,’;
            }
        }
        return c;
    }else if(count>1&& count<6){
        c+=‘are ’+count+‘ results: ’;
        for(var f=0; f<count; f++){
            c+=b.results.bindings[f].cittaLabel.value;
            if(f!=count-1){
                c+=‘,’;
            }
        }
        return c;
    }else if(count==1){
        c+=‘is ’;
        c+=b.results.bindings[0].cittaLabel.value;
        return c;
    }else if(count==0){
        c+=b.results.bindings[0].cittaLabel.value;
    }
    c+=b.results.bindings[0].cittaLabel.value;
    return c;
}
```

Questa funzione prende in input la lista di entità e la proprietà ottenute dalle funzioni getEntity e getProperty. Genera la query SPARQL per ottenere le informazioni e tramite una chiamata sincrona recupera l'oggetto JSON con le informazioni ottenute dalla query lanciata. Seguono una serie di controlli sulla lunghezza per verbalizzare la risposta in output. Anche in caso non vi sono risultati si restituisce il valore della label per attivare il meccanismo dell'Handler_error, in questo modo lo smart speaker dirà di non aver trovato alcun risultato.

3.8 Pubblicazione

A questo punto non resta altro che procedere con la "Distribution" cliccando sul relativo tab presente nella Alexa Developer Console, compilare tutti i campi obbligatori e procedere con il salvataggio. Seguirà un wizard che verificherà la correttezza di tutti i requisiti necessari controllando anche se vi sono conflitti tra utterances di vari intent, infine comunicherà eventuali correzioni da apportare. Quando tutto sarà corretto saremo indirizzati alla pagina Certification dalla quale potremo sottomettere la Skill che passerà in fase di Review. Dopo qualche giorno il team Amazon manderà un email allo sviluppatore, in caso di fallimento in essa vi sono contenuti suggerimenti per la risoluzione dei problemi che hanno causato il failure.

La skill è disponibile nello store inglese di Amazon cercando *Personal assistant based on Wikidata*.

3.8.1 Problema slot

Un problema riscontrato dopo la fase di review è stato un problema relativo agli slot. Amazon impone dei limiti per gli slot, ovvero per avere contenuto dinamico si dovrebbe utilizzare uno slot fornita da Amazon stessa, "Amazon.SearchQuery". Però questa possibilità restringe ad un unico intent e una sola utterances in quanto la frase prevede "ask, ⟨question⟩?". L'altra possibilità prevede creare una slot personalizzata con tutti i possibili valori delle slot. L'idea è utilizzando le query su Wikidata formulare delle query per ottenere i possibili valori delle slot scaricando il file .csv dei risultati trovati.

Però la quantità dei dati presenti nella Knowledge Graph è molto grande, quindi non è possibile caricare tutti i valori in quanto il modello di interazione può raggiungere una grandezza massima di 1.5MB. La soluzione trovata è effettuare query di tipologie diverse, per ottenere entità di svariato genere, e poi concatenare i file .csv di tutti i risultati ottenuti. Elenchiamo in seguito le query utilizzate per riempire gli slot "property" e "entity":

- Property: Query per recuperare tutte le entità associate a proprietà (5,913 risultati)

```
SELECT DISTINCT ?entity ?entityLabel WHERE {
    ?entity wdt:P1687 ?id
    SERVICE wikibase:label {
        bd:serviceParam wikibase:language "en".
    }
}
```

- Entity (in totale quasi 30,500 entità):
  - Query per recuperare persone:

```
SELECT DISTINCT ?entityLabel WHERE {
    ?entity wdt:P31 wd:Q5.
    SERVICE wikibase:label {
            bd:serviceParam wikibase:language "en".
    }
}LIMIT 10000
```

  - Query per recuperare le città:

```
SELECT DISTINCT ?entityLabel WHERE {
    ?entity wdt:P31 wd:Q515.
    SERVICE wikibase:label {
            bd:serviceParam wikibase:language "en".
    }
}LIMIT 5000
```

  - Query per recuperare i paesi (173 risultati):

```
SELECT DISTINCT ?entityLabel WHERE {
    ?entity wdt:P31 wd:Q6256.
    SERVICE wikibase:label {
            bd:serviceParam wikibase:language "en".
    }
}
```

  - Query per recuperare i film:

```
SELECT DISTINCT ?entityLabel WHERE {
    ?entity wdt:P31 wd:Q11424.
    SERVICE wikibase:label {
            bd:serviceParam wikibase:language "en".
    }
}LIMIT 3000
```

  - Query per recuperare i libri:

```
SELECT DISTINCT ?entityLabel WHERE {
    ?entity wdt:P31 wd:Q47461344.
    SERVICE wikibase:label {
            bd:serviceParam wikibase:language "en".
    }
}LIMIT 3000
```

  - Query per recuperare opere d'arte:

```
SELECT DISTINCT ?entityLabel WHERE {
    ?entity wdt:P31 wd:Q3305213.
    SERVICE wikibase:label {
            bd:serviceParam wikibase:language "en".
    }
}LIMIT 3000
```

– Query per recuperare monumenti e attrazioni turistiche (2,468 risultati):

```
SELECT DISTINCT ?entityLabel WHERE {
    ?entity wdt:P31 wd:Q570116.
    SERVICE wikibase:label {
            bd:serviceParam wikibase:language "en".
    }
}
```

– Query per recuperare i musei:

```
SELECT DISTINCT ?entityLabel WHERE {
    ?entity wdt:P31 wd:Q33506.
    SERVICE wikibase:label {
            bd:serviceParam wikibase:language "en".
    }
}LIMIT 3000
```

– Query per recuperare gli animali:

```
SELECT DISTINCT ?entityLabel WHERE {
    ?entity wdt:P31 wd:Q16521.
    SERVICE wikibase:label {
            bd:serviceParam wikibase:language "en".
    }
}LIMIT 1000
```

CAPITOLO 4

Testing

4.1 Valutazione

Per la valutazione della skill il metodo selezionato è stato quello di effettuare testing su un dataset challenge sui Linked Data, fornito da QALD, contenente domande risposte con Wikidata. Il datset include 100 domande fattuali, nella tabella seguente verranno messe a confronto queste domande con le domande formulate allo smart speaker per ottenere una risposta.

Tabella 4.1: Domande Dataset vs Domande Skill

id	**Domande Dataset**	**Domande Skill**
1	*Who was the doctoral supervisor of Albert Einstein?*	*Who is the doctoral supervisor of Albert Einstein?*
2	*Did Kaurismäki ever win the Grand Prix at Cannes?*	/
3	*Who wrote the song Hotel California?*	*Who is the composer of Hotel California?*
4	*Who was on the Apollo 11 mission?*	*Who is crew member of Apollo 11?*
5	*Which electronics companies were founded in Beijing?*	/
6	*What is in a chocolate chip cookie?*	*What is has part of chocolate chip cookie?*
7	*Was the wife of president Lincoln called Mary?*	*Who is the wife of president Lincoln?*
8	*How many movies did Park Chan-wook direct?*	*Which is the entity has Park Chan-Wook as director?*

9	*In which U.S. state is Mount McKinley located?*	*Where is mount McKinley?*
10	*Which Indian company has the most employees?*	*What is the business with maximum employee whose location is india?*
11	*In which school did Obama's wife study?*	*what is the has educated of Michelle Obama?*
12	*What is the capital of Cameroon?*	*What is the capital of Cameroon?*
13	*Where does Piccadilly start?*	*What is the terminus of Piccadilly?*
14	*When did the Boston Tea Party take place?*	*What is the point in time of Boston Tea Party?*
15	*Who played Gus Fring in Breaking Bad?*	*Who is the actor of Gus Fring?*
16	*Who wrote Harry Potter?*	*Who is the author of Harry Potter?*
17	*Which actors play in Big Bang Theory?*	*Who is the actor of Big Bang Theory?*
18	*What is the largest country in the world?*	*What is the country with maximum area?*
19	*Which states border Illinois?*	*What is the border of Illinois?*
20	*How many goals did Pelé score?*	/
21	*Who is the president of Eritrea?*	*Who is the president of Eritrea?*
22	*Which computer scientist won an oscar?*	/
23	*Who created Family Guy?*	*Who is the creator of Family Guy?*
24	*How many people live in Poland?*	*What is the population of Poland?*
25	*To which party does the mayor of Paris belong?*	*What is the political party of Anne Hidalgo?*
26	*Who does the voice of Bart Simpson?*	*Who is the various artists of Bart Simpson?*
27	*Who composed the soundtrack for Cameron's Titanic?*	*Who is the composer of Titanic?*
28	*When did Boris Becker end his active career?*	/
29	*Show me all basketball players that are higher than 2 meters.*	*Which person has the same occupation as basketball player and her height is more to 200?*
30	*What country is Sitecore from?*	*What is the country of Sitecore?*
31	*Which country was Bill Gates born in?*	*What is the citizenship of Bill Gates?*
32	*Who developed Slack?*	*Which is the entity has Slack HQ as product?*
33	*In which city did Nikos Kazantzakis die?*	*What is the place of death of Nikos Kazantzakis?*

34	*How many grand-children did Jacques Cousteau have?*	/
35	*Which films did Stanley Kubrick direct?*	*Which is the entity has Stanley Kubrick as director?*
36	*Does Neymar play for Real Madrid?*	*What is the sports team of Neymar?*
37	*How many seats does the home stadium of FC Porto have?*	*What is the seating capacity of Estádio do Dragão?*
38	*Show me all books in Asimov's Foundation series.*	*What is the has part of Foundation series?*
39	*Which movies star both Liz Taylor and Richard Burton?*	/
40	*In which city are the headquarters of the United Nations?*	*Where is the United Nations headquarters?*
41	*In which city was the president of Montenegro born?*	*What is the birthplace of Milo Dukanović?*
42	*Which writers studied in Istanbul?*	/
43	*Who is the mayor of Paris?*	*Who is the head of government of Paris?*
44	*What is the full name of Prince Charles?*	*What is the given name of Prince Charles?*
45	*What is the longest river in China?*	*What is the river with maximum length whose location is China?*
46	*Who discovered Ceres?*	*Who is the discoverer of Ceres?*
47	*When did princess Diana die?*	*When is the date of death of princess Diana?*
48	*Which presidents were born in 1945?*	*Which person has the same position held as president and her date of birth is equal to 1945?*
49	*Give me all federal chancellors of Germany.*	*Which is the entity has federal chancellor of Germany as position held?*
50	*How many moons does Mars have?*	*Who is the orbiting body of Mars?*
51	*What was the first Queen album?*	/
52	*Did Elvis Presley have children?*	*Who is the child of Elvis Presley?*
53	*Give me a list of all Canadians that reside in the U.S.*	/
54	*Where is Syngman Rhee buried?*	*What is the place of burial of Rhee Syng-Man?*
55	*In which countries do people speak Japanese?*	*What country has official language like Japanese?*
56	*Who is the king of the Netherlands?*	*Who is the governor of Netherlands?*

57	*Who produced the most films?*	/
58	*Show me all Czech movies.*	*What film has country of origin like Czech Republic? + What film has country of origin like Czechoslovakia?*
59	*Which rivers flow into the North Sea?*	*What is the tributary of North Sea?*
60	*When did Operation Overlord commence?*	*When is the start time of operation Overlord?*
61	*Where do the Red Sox play?*	*What is the stadium of Red Sox?*
62	*In which time zone is Rome?*	/
63	*Give me a list of all critically endangered birds.*	*What bird has conservation status like critically endangered?*
64	*How much did the Lego Movie cost?*	*What is the cost of Lego Movie?*
65	*What was the original occupation of the inventor of Lego?*	*Who is the inventor of Lego? + What is the occupation of Ole Kirk Christiansen?*
66	*Which countries have more than ten volcanoes?*	/
67	*Show me all U.S. states.*	*Which is the entity has US state as instance of?*
68	*Who wrote the Game of Thrones theme?*	*What is the composer of Game of Thrones?*
69	*Give me all films produced by Hal Roach.*	*What film has producer like Hal Roach?*
70	*In which films did Julia Roberts as well as Richard Gere play?*	/
71	*When was Olof Palme shot?*	*When is the date of death of Olof Palme?*
72	*How many emperors did China have?*	*Which is the entity has emperor of China as position held?*
73	*Show me the book that Muhammad Ali wrote.*	*What literary work has author like Muhammad Ali?*
74	*Which country has the most official languages?*	*What is the stadium with maximum seating capacity whose location is Spain?*
75	*How did Michael Jackson die?*	*What is the cause of death of Michael Jackson?*
76	*Which space probes were sent into orbit around the sun?*	/
77	*Who produced films starring Natalie Portman?*	/
78	*What is the biggest stadium in Spain?*	*What is the most seating capacity of stadium whose location is Spain?*

79	*On which day is Columbus Day?*	*What is the recurrence of Columbus Day?*
80	*How short is the shortest active NBA player?*	/
81	*Whom did Lance Bass marry?*	*Who is the wife of Lance Bass?*
82	*What form of government does Russia have?*	*What is the form of government of Russia?*
83	*What movies does Jesse Eisenberg play in?*	*Which is the entity has Jesse Eisenberg as cast member?*
84	*Give me all soccer clubs in the Premier League.*	*What association football club has league like Premier League?*
85	*Show me all museums in London.*	*What museum has relative position like London?*
86	*Give me all South American countries.*	*What country has part like South America?*
87	*Which pope succeeded John Paul II?*	/
88	*Who is the son of Sonny and Cher?*	*Who is the child of Sonny Bono? + Who is the child of Cher?*
89	*What are the five boroughs of New York?*	*Which is the entity has borough of New York City as instance of?*
90	*Show me Hemingway's autobiography.*	/
91	*What kind of music did Lou Reed play?*	*What is the genre of Lou Reed?*
92	*In which city does Sylvester Stallone live?*	*What is the residence of Sylvester Stallone?*
93	*Who was Vincent van Gogh inspired by?*	*What is the influence of Vincent Van Gogh?*
94	*What are the names of the Teenage Mutant Ninja Turtles?*	*What is the has part of TMNT?*
95	*What are the zodiac signs?*	*What is the has part of zodiac?*
96	*What languages do they speak in Pakistan?*	*What is the official language of Pakistan?*
97	*Who became president after JFK died?*	/
98	*In what city is the Heineken brewery?*	*Where is Heineken brewery?*
99	*What is Elon Musk famous for?*	/
100	*What is Batman's real name?*	*What is the birth name of Batman?*

Nella tabella 4.1 sono presenti domande che la nostra skill non è riuscita a rispondere, segnati con "/".

Infine esaminiamo le risposte del dataset confrontandole con le risposte ottenute dalla skill, per analizzare la validità di quest'ultima.

Tabella 4.2: Risposte Dataset vs Risposte Skill

id	Domande Dataset	Domande Skill
1	*Heinrich Burkhardt, Heinrich Friedrich Weber, Alfred Kleiner*	*The doctoral supervisor of Albert Einstein are 3 results: Heinrich Burkhardt, Heinrich Friedrich Weber, Alfred Kleiner. Do you have more questions?*
2	*True*	/
3	*Don Felder*	*The composer of Hotel California is Don Felder. Do you have more questions?*
4	*Neil Armstrong, Buzz Aldrin, Michael Collins*	*The crew member of Apollo 11 are 3 results: Neil Armstrong, Buzz Aldrin, Michael Collins. Do you have more questions?*
5	*Xiaomi*	/
6	*flour, chocolate chip*	*The has part of chocolate chip cookie are 2 results: flour, chocolate chip. Do you have more questions?*
7	*Mary Todd Lincoln*	*The wife of president Lincoln is Mary Todd Lincoln. Do you have more questions?*
8	*17*	*The director of Park Chan-Wook are 17 results, here are the top 5: Joint Security Area, Oldboy, The Vengeance Trilogy, Thirst, Sympathy for Mr. Vengeance. Do you have more questions?*
9	*Denali*	*The mount McKinley is in: Denali National Park and Preserve, Denali Borough, United States of America. Do you have more questions?*
10	*Indian Railways*	*The maximum employee of business in India is Indian Railways. Do you have more questions?*
11	*Princeton University, Harvard Law School, Whitney M. Young Magnet High School*	*The has educated of Michelle Obama are 3 results: Princeton University, Harvard Law School, Whitney M. Young Magnet High School. Do you have more questions?*
12	*Yaoundé*	*The capital of Cameroon is Yaoundé. Do you have more questions?*

13	*Piccadilly Circus, Hyde Park Corner*	*The terminus of Piccadilly are 2 results: Piccadilly Circus, Hyde Park Corner. Do you have more questions?*
14	*1773-12-16T00:00:00Z*	*The point in time of Boston Tea Party is 1773-12-16T00:00:00Z. Do you have more questions?*
15	*Giancarlo Esposito*	*The actor of Gus Fring is Giancarlo Esposito. Do you have more questions?*
16	*J. K. Rowling*	*The author of Harry Potter is J. K. Rowling. Do you have more questions?*
17	*(58 risultati) Kaley Cuoco, Charlie Sheen, George Takei, Riki Lindhome, Stan Lee,...*	*The actor of Big Bang Theory are 58 results, here are the top 5: Kaley Cuoco, Charlie Sheen, George Takei, Riki Lindhome, Stan Lee. Do you have more questions?*
18	*Canada*	*The most maximum of country is Canada. Do you have more questions?*
19	*Michigan, Indiana, Wisconsin, Iowa, Missouri, Kentucky*	*The border of Illinois are 6 results, here are the top 5: Michigan,Indiana,Wisconsin,Iowa,Missouri. Do you have more questions?*
20	*679*	/
21	*Isaias Afwerki*	*The president of Eritrea is Isaias Afwerki. Do you have more questions?*
22	*Paul Debevec, Edwin Catmull, Demetri Terzopoulos, Henrik Wann Jensen*	/
23	*Seth MacFarlane*	*The creator of Family Guy is Seth MacFarlane. Do you have more questions?*
24	*38454576*	*The population of Poland is 38454576. Do you have more questions?*
25	*Socialist Party*	*The political party of Anne Hidalgo is Socialist Party. Do you have more questions?*
26	*Nancy Cartwright*	*The various artists of Bart Simpson are 2 results: Nancy Cartwright, Nathalie Bienaimé. Do you have more questions?*
27	*James Horner*	*The composer of Titanic is James Horner. Do you have more questions?*
28	*1999-01-01T00:00:00Z*	/

29	*(5194 risultati) Thaddeus Young, Coleman Collins, Maurice Harkless, Georgi Glouchkov, Detlef Schrempf,...*	*The occupation of basketball player height more 200 are 5194 results, here are the top 5: Thaddeus Young, Coleman Collins, Maurice Harkless, Georgi Glouchkov, Detlef Schrempf. Do you have more questions?*
30	*Denmark*	*The country of Sitecore is Denmark. Do you have more questions?*
31	*United States of America*	*The citizenship of Bill Gates is United States of America. Do you have more questions?*
32	*Slack Technologies*	*The product of Slack HQ is Slack Technologies. Do you have more questions?*
33	*Freiburg im Breisgau*	*The place of death of Nikos Kazantzakis is Freiburg im Breisgau. Do you have more questions?*
34	*4*	/
35	*(18 risultati) A Space Odyssey, Dr. Strangelove, Spartacus, A Clockwork Orange, The Shining,...*	*The director of Stanley Kubrick are 18 results, here are the top 5: 2001: A Space Odyssey, Dr. Strangelove, Spartacus, A Clockwork Orange, The Shining. Do you have more questions?*
36	*Paris Saint-Germain F.C.*	*The sports team of Neymar is Paris Saint-Germain F.C.. Do you have more questions?*
37	*50431*	*The seating capacity of Estádio do Dragão is 50431. Do you have more questions?*
38	*(8 risultati) Foundation, Prelude to Foundation, Foundation and Empire, Second Foundation, Robot short stories,...*	*The has part of Foundation series are 8 results, here are the top 5: Foundation, Prelude to Foundation, Foundation and Empire, Second Foundation, Robot short stories. Do you have more questions?*
39	*12 risultati*	/
40	/	*The united nations headquarters is in: Manhattan, Turtle Bay, United States of America. Do you have more questions?*
41	*Nikšić*	*The birthplace of Milo Dukanović is Nikšić. Do you have more questions?*
42	*191 risultati*	/

43	*Anne Hidalgo*	*The head of government of Paris is Anne Hidalgo. Do you have more questions?*
44	*(226 risultati) Charles, Prince of Wales in tutte le lingue del mondo*	*The given name of Prince Charles is Charles, Prince of Wales. Do you have more questions?*
45	*Yangtze*	*The maximum length of river in China is Yangtze. Do you have more questions?*
46	*Giuseppe Piazzi*	*The discoverer of Ceres is Giuseppe Piazzi. Do you have more questions?*
47	*1997-08-31T00:00:00Z*	*The date of death of princess Diana is 1997-08-31T00:00:00Z. Do you have more questions?*
48	*(42 risultati) Angus Deaton, Richard Thaler, Alain Robert, Jean-Marie Delarue, Yannick Moreau,...*	*The position held of president date of birth equal 1945 are 31 results, here are the top 5: Angus Deaton, Richard Thaler, Alain Robert, Jean-Marie Delarue, Yannick Moreau. Do you have more questions?*
49	*(9 risultati) Angela Merkel, Konrad Adenauer, Ludwig Erhard, Kurt Georg Kiesinger, Willy Brandt,...*	*The position held of federal chancellor of Germany are 9 results, here are the top 5: Angela Merkel, Konrad Adenauer, Ludwig Erhard, Kurt Georg Kiesinger, Willy Brandt. Do you have more questions?*
50	*Phobos, Deimos*	*The orbiting body of mars are 2 results: Phobos, Deimos. Do you have more questions?*
51	*Live Killers*	/
52	*True*	*The child of Elvis Presley is Lisa Marie Presley. Do you have more questions?*
53	*James Hathaway, Karl W. Giberson, Pierre-Gabriel Huot*	/
54	*Seoul National Cemetery*	*The place of burial of Rhee Syng-Man is Seoul National Cemetery. Do you have more questions?*
55	*Japan*	*The country with official language by Japanese is Japan. Do you have more questions?*

56	*Willem-Alexander of the Netherlands*	*The governor of Netherlands is Willem-Alexander of the Netherlands. Do you have more questions?*
57	*Jesse Louis Lasky*	/
58	*1512 risultati*	*The film with country of origin by Czech republic are 650 results, here are the top 5: Kooky, Not Angels But Angels, Pánská jízda, Divided We Fall, Želary. Do you have more questions? + The film with country of origin by Czechoslovakia are 862 results, here are the top 5: Babičky dobíjejte přesně!, Adventure in Golden Bay, Boris Godunov, Day for My Love, Courage for Every Day. Do you have more questions?*
59	*(89 risultati) Rhine, Elbe, Weser, Eider, River Thames,...*	*The tributary of North Sea are 89 results, here are the top 5: Rhine, Elbe, Weser, Eider, River Thames. Do you have more questions?*
60	*1944-06-06T00:00:00Z*	*The start time of operation Overlord is 1944-06-06T00:00:00Z. Do you have more questions?*
61	*Fenway Park*	*The stadium of Red Sox is Fenway Park. Do you have more questions?*
62	/	/
63	*224 risultati*	*The bird with conservation status by critically endangered is Stresemann's Bristlefront. Do you have more questions?*
64	*60000000*	*The cost of Lego Movie is 60000000. Do you have more questions?*
65	*entrepreneur, carpenter, designer, organizational founder*	*The inventor of Lego is Ole Kirk Christiansen. Do you have more questions? + The occupation of Ole Kirk Christiansen is entrepreneur, carpenter, designer, organizational founder. Do you have more questions?*
66	*40 risultati*	/

67	*(50 risultati) California, Alabama, Maine, New Hampshire, Massachusetts,...*	*The instance of of US State are 50 results, here are the top 5: California, Alabama, Maine, New Hampshire, Massachusetts. Do you have more questions?*
68	*Ramin Djawadi*	*The composer of Game of Thrones is Ramin Djawadi. Do you have more questions?*
69	*(209 risultati) Number, Please?, A Sailor-Made Man, Why Worry?, Bad Boy, Bungalow Boobs,...*	*The film with producer by Hal roach are 209 results, here are the top 5: Number, Please?, A Sailor-Made Man, Why Worry?, Bad Boy, Bungalow Boobs. Do you have more questions?*
70	*Pretty Woman , Runaway Bride*	/
71	*1986-02-28T00:00:00Z*	*The date of death of Olof Palme is 1986-02-28T00:00:00Z. Do you have more questions?*
72	*330*	*The position held of emperor of china are 330 results, here are the top 5: Emperor Shizong of Liao, Emperor Muzong of Liao, Emperor Jingzong of Liao, Emperor Shengzong of Liao, Emperor Xingzong of Liao. Do you have more questions?*
73	*The Soul of a Butterfly, The Greatest: My Own Story*	*The literary work with author by Muhammad Ali are 2 results: The Soul of a Butterfly, The Greatest: My Own Story. Do you have more questions?*
74	*Zimbabwe*	*The maximum official language of country is Bangladesh. Do you have more questions?*
75	*myocardial infarction, combined drug intoxication*	*The cause of death of Michael Jackson are 2 results: myocardial infarction, combined drug intoxication. Do you have more questions?*
76	*8 risultati*	/
77	*59 risultati*	/
78	*Estadi Olímpic Lluís Companys*	*The maximum seating capacity of stadium in Spain is Estadi Olímpic Lluís Companys. Do you have more questions?*

79	*second Monday in October*	*The recurrence of Columbus Day is second Monday in October. Do you have more questions?*
80	*1,85*	/
81	*Michael Turchin*	*The wife of Lance Bass is Michael Turchin. Do you have more questions?*
82	*Parliamentary Republic, representative democracy, federal republic*	*The form of government of Russia are 3 results: Parliamentary Republic, representative democracy, federal republic. Do you have more questions?*
83	*(35 risultati) To Rome with Love, The Social Network, Zombieland, 30 Minutes or Less, The Emperor's Club,...*	*The cast member of Jesse Eisenberg are 35 results, here are the top 5: To Rome with Love, The Social Network, Zombieland, 30 Minutes or Less, The Emperor's Club. Do you have more questions?*
84	*(22 risultati) Watford F.C., Everton F.C., Chelsea F.C., Arsenal F.C., Manchester United F.C.,...*	*The association football club with league by premier league are 22 results, here are the top 5: Watford F.C., Everton F.C., Chelsea F.C., Arsenal F.C., Manchester United F.C.. Do you have more questions?*
85	*V and A Museum of Childhood, London amora, Petrie Museum of Egyptian Archaeology, Bramah Tea and Coffee Museum*	*The museum with relative position by London are 4 results: V and A Museum of Childhood, London amora, Petrie Museum of Egyptian Archaeology, Bramah Tea and Coffee Museum. Do you have more questions?*
86	*(11 risultati) Uruguay, Brazil, Chile, Argentina, Peru,...*	*The country with part by south America are 11 results, here are the top 5: Uruguay, Brazil, Chile, Argentina, Peru. Do you have more questions?*
87	*Benedict XVI*	/
88	*Chaz Bono*	*The child of sonny Bono is Chaz Bono. Do you have more questions? + The child of Cher are 2 results: Chaz Bono, Elijah Blue Allman. Do you have more questions?*
89	*Manhattan, Brooklyn, Queens, The Bronx, Staten Island*	*The instance of of borough of New York city are 5 results: Manhattan, Brooklyn, Queens, The Bronx, Staten Island. Do you have more questions?*

90	*A Moveable Feast*	/
91	*(12 risultati) alternative rock, rock music, protopunk, glam rock, noise rock,...*	*The genre of Lou reed are 12 results, here are the top 5: alternative rock, rock music, protopunk, glam rock, noise rock. Do you have more questions?*
92	*Beverly Hills*	*The residence of Sylvester Stallone is Beverly Hills. Do you have more questions?*
93	*Peter Paul Rubens, Paul Cézanne, Paul Gauguin, Jean-François Millet, Anton Mauve, Willem Roelofs*	*The influence of Vincent van Gogh are 6 results, here are the top 5: Peter Paul Rubens, Paul Cézanne, Paul Gauguin, Jean-François Millet, Anton Mauve. Do you have more questions?*
94	*Michelangelo, Donatello, Leonardo, Raphael*	*The has part of tmnt are 4 results: Michelangelo, Donatello, Leonardo, Raphael. Do you have more questions?*
95	*(12 risultati) Aries, Gemini, Virgo, Libra, Scorpio*	*The has part of zodiac are 12 results, here are the top 5: Aries, Gemini, Virgo, Libra, Scorpio. Do you have more questions?*
96	*Urdu, English*	*The official language of Pakistan are 2 results: Urdu, English. Do you have more questions?*
97	*Lyndon B. Johnson*	/
98	*Amsterdam*	*The Heineken brewery is in: Stadhouderskade 78-79, Amsterdam, De Pijp, Netherlands. Do you have more questions?*
99	/	/
100	*Bruce Wayne*	*The birth name of Batman is Bruce Wayne. Do you have more questions?*

4.2 Risultati

Esaminiamo le domande non coperte o coperte parzialmente dalla nostra skill.

- Le domande 2 e 52 non utilizzavano query modellate con SELECT ma con ASK, cioè come risposta era richiesto un valore TRUE o FALSE. Per la domanda 52 tramite una SELECT si riesce ad ottenere comunque una risposta alla domanda, trasformandola da "Did Elvis Presley have children?" a "Who is the child of

Elvis Presley?". Per quanto riguarda la domanda 2 non è stato possibili ottenere alcuna risposta.

- Le domande 5, 22 e 90 dovrebbero ricadere nel pattern "getResultFilter" ma la proprietà per modellare le query dovrebbe essere diversa da "instance of", cioè si dovrebbe aggiungere un ulteriore parametro nella utterance, passando da "what ⟨entity⟩ has ⟨property⟩ like ⟨filter⟩?" a "what ⟨property⟩ of ⟨entity⟩ has ⟨property2⟩ like ⟨filter⟩?". L'utente dovrebbe però sapere quando usare "instance of" e quando un'altra proprietà. Nel caso della domanda 5 la proprietà è "industry", nella domanda 22 è "occupation" e nella 90 è "genre". Inoltre anche modificando il pattern, nello specifico la domanda 22 non verrebbe comunque risposta in quanto per ottenere la risposta desiderata servirebbe un ulteriore predicato.

- Le domande 20, 87 e 97 usano dei qualificatori di Wikidata, attributi che qualificano caratteristiche della coppia (proprietà, valore).

- Le domande 34, 42, 48, 53, 63 e 80 utilizzano degli operatori come "/" o "*" per usare più predicati nella stessa tripla, ad esempio "?uri wdt:P19/wdt:P17 wd:Q16.". Nelle domande 48 e 63 si riesce ad ottenere una risposta parziale utilizzando un predicato. Per la domanda 80 oltre al qualificatore per ottenere una risposta viene usato anche il FILTER NOT EXISTS.

- Le domande 28 e 76 non riusciamo ad ottenere una risposta perchè le entità associate alle wikidata property sono 2. Purtroppo non è possibile applicarle a cascata poichè spesso restituiscono un valore in risposta anche con la proprietà che riteniamo errata.

- Domande 39, 66 e 70 sono domande il quale pattern identificato non è supportato attualmente, cioè trovare risultati con la stessa proprietà ma con 2 entità differenti, ad esempio "Which movies star both Liz Taylor and Richard Burton?" diventerebbe "Which film has Liz Taylor like cast member and Richard Burton like cast member?".

- Domande 40, 62 e 99 non ottengono risultati neanche nel dataset poichè dopo alcuni aggiornamenti in Wikidata le proprietà e i risultati richiesti non sono più presenti nelle entità. Nella domanda 40 riusciamo ad ottenere una risposta diversa ma concettualmente simile a quella richiesta.

- Domanda 44 la domanda risposta dalla skill è quella ritenuta più adeguata alla domanda del dataset rispetto ai risultati attesi dal dataset in cui vi è solo un elenco dello stesso nome in varie lingue del mondo.

- Domanda 51 è un pattern non supportato. In cui si potrebbe generalizzare con domande first/last, cioè "What was the first/last Queen album?".

- Domanda 57, 74 dovrebbero essere risposte dal pattern superlative ma la struttura query è leggermente diversa in quanto sui risultati viene appliccato un GROUP BY e un COUNT sull'unità di misura per la comparazione. Una soluzione potrebbe essere quella di mettere a cascata la struttura identificata precedentemente e questa identificata. Però anche in questo caso non otterrebbe comunque il risultato giusto in quanto si fermerebbero alla prima struttura con un risultato diverso da quello atteso.

- Domanda 66 ha una struttura della query ad hoc. in quanto per poterla rispondere occorrerebbe una UNION un COUNT,un GROUP BY e un HAVING. In conclusione non generalizzabile.

Alla fine della valutazione la skill risponde correttamente a 73 domande, 2 domande con risposte ritenute diverse ma contestualmente corrette, 3 domande impossibili da rispondere, in quanto dopo l'aggiornamento del knowledge graph le informazioni richieste non erano più presenti, mentre le restanti 22 domande non risposte o con risposte parziali per pattern non supportati.

CAPITOLO 5

Conclusioni e sviluppi futuri

Il risultato finale è quindi una skill che data in input una domanda dall'utente, definita come una delle utterance del modello di interazione del paragrafo 3.6, identifica le parole chiave come slot passandole al back-end. Il back-end, cioè la Lambda function in AWS, recupera gli URI di queste parole tramite le API di Wikidata e in seguito viene creata dinamicamente, a seconda dell'intent invocato, una query SPARQL con gli URI ottenuti che ci fornisce le informazioni cercate, infine queste informazioni vengono date in output dal dispositivo vocale.

5.1 Sviluppi Futuri

Attualmente la skill per ovviarre al problema degli slot (discusso nel sottoparagrafo 3.8.1) è costretto ad avere nello slot tutti i valori possibili, ma questo è una grande limitazione in quanto è possibile ottenere solo un sottoinsieme delle informazioni contenute nella KG. Come implementazione futura si potrebbe utilizzare lo slot dinamico fornito da AMAZON ma la logica per riconoscere le frasi e le parole dovrebbero essere gestite nel back-end e questo comporterebbe a perdere il meccanismo di riconoscimento fornito dalla Alexa Developer Console. Attualmente si è cercato di fornire risposte a domande relative ad ambiti eterogenei. Per questo è stato recuperato un numero limitato di valori per diversi argomenti trattati da Wikidata (persone, luoghi, animali, opere d'arte). L'alternativa sarebbe focalizzarsi su un argomento, ad esempio beni culturali, e recuperare tutti i dati che permettono di rispondere...

Dopo la valutazione sul dataset, nel paragrafo 4.2, sono stati identificati nuovi pattern (query con più di un predicato, domande con first/last, ecc). Uno sviluppo futuro potrebbe essere l'implementazione di quest'ultimi, però da valutare il focus da dare alla futura skill in quanto il modello di interazione deve rientrare nei 1.5 MB, quindi aggiungendo nuovi intent, che invocano i nuovi pattern, dovremmo sacrificare delle entity inserite nello slot.

La skill pubblicata è stata sviluppata in sola lingua inglese. Potrebbe essere interessante sviluppare la skill anche in altre lingue, compreso l'italiano.

Altro sviluppo possibile in futuro è l'implementazione di una skill in grado di interrogare altre KG (come DBpedia) o più KG contemporaneamente. Magari al termine dell'esecuzione delle query fare un matching delle risposte ottenute dalle interrogazioni sui diversi KG, se sono uguali allora la risposta è corretta al 100% altrimenti se ha valori diversi da il primo trovato avvisando dell'ambiguità riscontrata.

5.1.1 DBpedia

Sulla scia di quanto detto andiamo ad analizzare il percorso per sviluppare una skill con una KG diversa, cioè DBpedia, verifichiamo le API disponibili per questa KG nella libreria dbpedia-entity-lookup per NodeJS. Le funzioni disponibili sono: findPerson(query), findPlace(query), findOrganization(query), findTitle(query), dove l'argomento query è un oggetto:

```
{
    entity:  "The name of the thing the user wants to find."
    options: TBD
}
```

facendo un esempio con l'entità "Paris", bisogna utilizzare la funzione per trovare i luoghi findPlace(query), con la variabile "query" definita nel seguente modo:

```
{
    entity:  "Paris".
}
```

l'oggetto in risposta risulterebbe il seguente:

```
{
   Object
   description: "Paris is the capital and largest city of France.
       It is situated on the river Seine, in northern France, at
      the heart of the Ile-de-France"
```

```
    id: "http://dbpedia.org/resource/Paris"
    name: "Paris"
    nameType: "place"
    originalQueryString: "paris"
    repository: "dbpedia"
    uri: "http://dbpedia.org/resource/Paris"
    uriForDisplay: "https://dbpedia.lookup.services.cwrc.ca/
        resource/Paris"
}
```

Non c'è un modo, tramite API, di recuperare le proprietà.

Per poter effettuare la query si utilizzerebbe il seguente modulo:

```
var dps = require('dbpedia-sparql-client').default;
var query = 'SELECT DISTINCT ?Concept WHERE {[] a ?Concept} LIMIT 10';

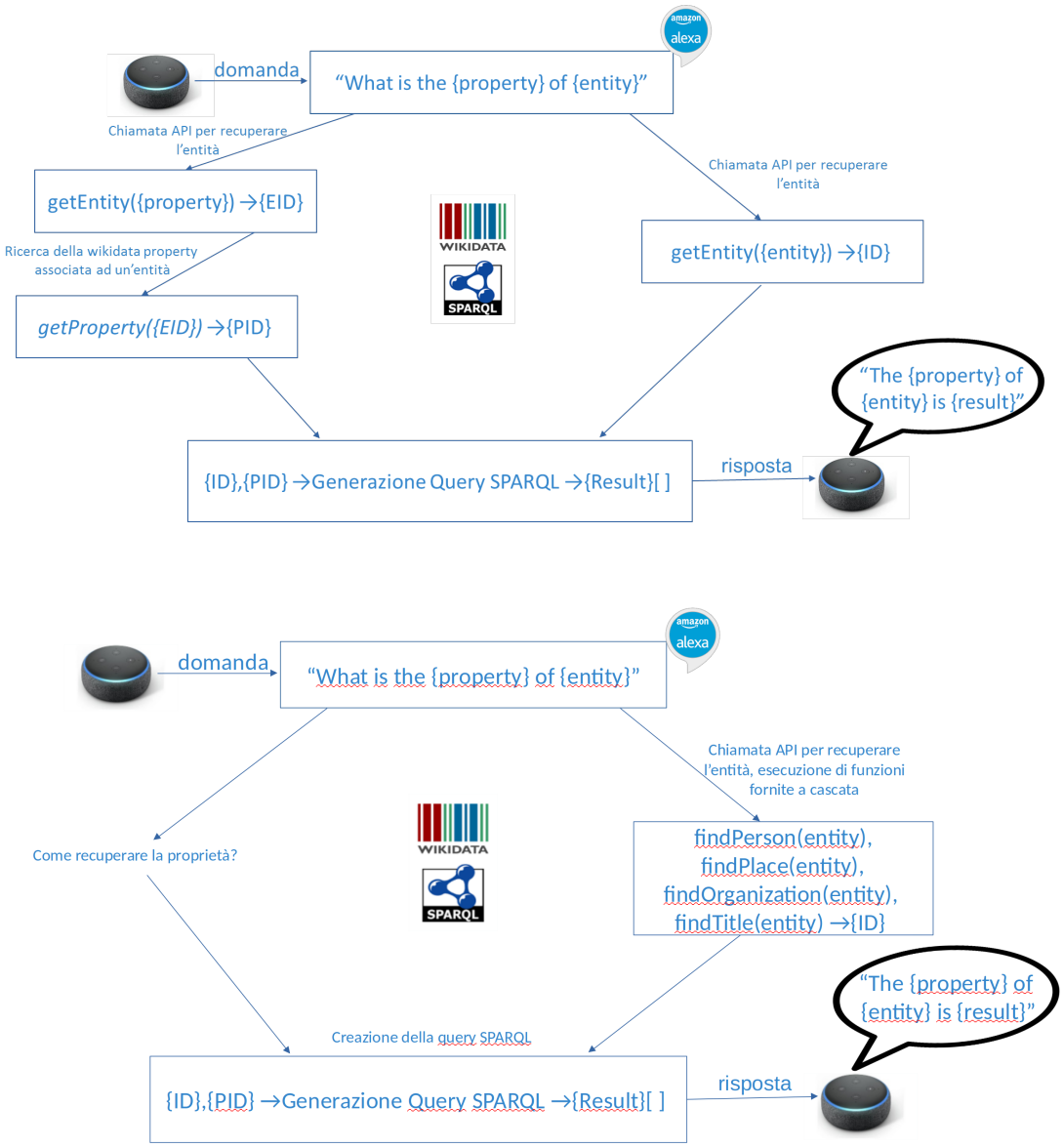
dps.client()
  .query(query)
  .timeout(15000) // optional, defaults to 10000
  .asJson() // or asXml()
  .then(function(r) { /* handle success */ })
  .catch(function(e) { /* handle error */ });
```

La struttura della query di DBpedia è un po' diversa da quelle viste in Wikidata, in quanto per ottenere le label non è previsto l'utilizzo di SERVICE.

Vediamo a confronto i due workflow per la skill in Wikidata e con la skill in DBpedia

- **Wikidata**: utilizzo di API per recuperare qualsiasi entità, utilizzo di API per recuperare l'entità (associata alla proprietà) e tramite query dinamica recupero l'effettiva proprietà legata all'entità (wikidata property associata). Trovati i due valori (entità e proprietà) creo dinamicamente la query.
- **DBpedia**: utilizzare API per recuperare l'id dell'entità però solo se essa ricade nelle risorse Luoghi, Organizzazioni, Titoli e Persone tramite le funzioni findPerson(query), findPlace(query), findOrganization(query), findTitle(query) dunque non sapendo quale entità utilizzerà l'utente bisognerebbe provare a cascata tutte le 4 funzioni fornite. Mentre non essendoci una funzione per recuperare la proprietà e non essendoci un'associazione tra le entità e proprietà in DBpedia non è possibile recuperare questo valore.

Nelle figure successive verrà schematizzato prima il workflow di Wikidata e poi il workflow di DBpedia.



Dunque per poter implementare una skill con DBpedia come KG occorre trovare un modo per recuperare le proprietà e le entità, mentre le query implementate si prestano ad essere generalizzate.

Ringraziamenti

Vorrei innanzitutto ringraziare il mio relatore, il professor Vittorio Scarano, per avermi accettato come tesista e per il bellissimo lavoro affidatomi, e la dottoressa Maria Angela Pellegrino, per il suo aiuto, la sua pazienza e per avermi trasmesso interesse nel campo del Semantic Web. In generale voglio ringraziare tutto l'ISISLab, perchè è un'ambiente bellissimo e gradevole, dall'organizzazione impeccabile. La festa di Natale e la pasta del professor Scarano rimarranno ricordi preziosi.

Ringrazio i miei genitori, a cui devo tutto, loro sono sempre stati lì ad incoraggiarmi in qualunque scelta, ad aiutarmi, a rialzarmi quando ne avevo bisogno, a spronarmi e consigliarmi, ma soprattutto hanno sempre creduto in me.

Ringraziamento speciale a Carmen, l'amore della mia vita, il mio sostegno, la mia forza, la mia confidente da oltre 10 anni. Abbiamo iniziato questo percorso insieme e senza di te forse non lo avrei neanche imboccato questo percorso che ora mi riempie di gioia e di orgoglio. Grazie per ogni istante in cui mi hai supportato e sopportato, dedico a te e ai miei genitori questo e i futuri successi.

Ringrazio i miei amici di sempre, Alessandro, Manuel, Angioletto, Andrea e mio cugino Alfonso perchè sono una costante su cui posso sempre contare.

Ringrazio gli amici fatti in questi 3 anni e mezzo di studi, con cui ho condiviso gioie e dolori, oltre che la passione per l'informatica. Li ringrazio per avermi aiutato, motivato e reso le cose divertenti anche nei momenti peggiori. In particolare ringrazio Rafiluccio, il grande amico di questa avventura, Matteo, Silvio, Francesca, Simone, Erminio, Paolo, Davide, Danilo, Angelo, Gennaro, Emanuele, Enzo e Pino. Anche se le

nostre strade dovessero dividersi i ricordi delle lunghe tavolate a mensa e le chiacchiere vicino la cattedra tra una lezione e un'altra rimarranno con me.

Ringrazio me stesso, perchè ci ho messo passione e impegno costantemente, nonostante ho iniziato questo percorso più tardi, ed oggi posso dire di essere fiero di me e di questo traguardo raggiunto.

E infine ringrazio l'informatica perchè mi ha donato un futuro con lei, dove potrò lavorare a cose che mi piacciono e che mi appassionano.

Bibliografia

[1] Amazon. Interaction model, 2020. Disponibile online.

[2] Amazon. Lambda function, 2020. Disponibile online.

[3] C. Bizer M. Schmachtenberg and H. Paulheim. Adoption of the linked data best practices in different topical domains. Spinger: in International Semantic Web Conference, (ISWC), pages 245–260, 2014.

[4] maxlath. Wikibase-sdk, 2020. Disponibile online.

[5] A. Polleres P. A. Bonatti, S. Decker and V. Presutti. Knowledge graphs: New directions for knowledge representation on the semantic web. Dagstuhl Reports, 2018.

[6] Wikidata. Sparql, 2020. Disponibile online.

[7] Wikipedia. Knowledge base, 2020. Disponibile online.

[8] Wikipedia. le 5 w, 2020. Disponibile online.

[9] Wikipedia. Sparql, 2020. Disponibile online.

[10] Wikipedia. Wikidata, 2020. Disponibile online.