**Theatro: Un Chatbot de pel·lícules i series implementat amb DialogFlow , Neo4J i GraphQL**

Biel Carpi · Victor Xirau · Rafael Morera

*Els ChatBots de conversa contemporanis són fàcils d'utilitzar i tenen la capacitat de simular converses humanes. Tanmateix, no poden avaluar grans conjunts de dades exhaustius per donar una resposta a l'usuari. En canvi, disposem de moltes APIs gratuïtes que contenen aquestes dades i poden oferir-les donat un seguit de paràmetres. En esta investigació vam dissenyar un servidor capaç de escoltar peticions de DialogFlow i processar-les emprant una base de dades de Graph per millorar l’experiència d’usuari i oferir un volum de informació superior al habitual al client. Mitjançant Google Dialogflow analitzem i GraphQL tradueix la query Cypher per peticionar la nostre base de dades Neo4J, aquesta analitza i proporciona una resposta precisa mitjançant un conjunt de dades complet i simula una conversa sembla a un humà. Utilitzem la base de dades per oferir respostes precises mitjançant un conjunt de dades de resposta i Google Dialogflow per simular interaccions humanes.*

*Termes Generals: chatbot, IA, movies, conversation, dialogflow, movies, tv-shows*

# Introducció

Tots nosaltres ens hem trobat en la situació d’estar navegant durant hores per Netflix, HBO, Disney+ cercant una pel·lícula o sèrie per veure, sense èxit. En quant es vol cercar informació sobre pel·lícules, sobre els actors o fer recomanacions informades, hem de passar a fer ús de pàgines com IMDB o Rotten Tomatoes, que son bastant complicades de emprar per un usuari mig. La nostres solució és simple: crear un chatbot que et recomana pel·lícules i series i proveeix informació sobre elles al usuari. Aquest es presentarà a través de la interfície d’un mòbil i es dirà Theatro.

Els ChatBots son programes que estimulen parlar com un humà. Sovint estan dissenyats amb un objectiu en ment – en el nostre cas, donar informació sobre contingut audiovisual ja siguin pel·lícules o series. Però és important saber que hi ha un seguit de components necessaris per construir un ChatBot:

1. *Processat de Llenguatge Natural* – separar les frases de input entre les parts que la formen
2. *Generació de Quèries –* traduir les frases processades a quèries valides
3. *Gestió de Context –* Recordar les respostes a quèries anteriors per contestar-ne futures de manera intel·ligent
4. *Coneixement del Domini –* Desar el coneixement sobre el domini en que el ChatBot se centra
5. *Generació de Llenguatge Natural –* Donar personalitat al ChatBot

Trobar la bona combinació d’eines que permetin construir un bon chatbot pot suposar un repte. Clarament existeixen moltes plataformes basades en Chatbots. Aquestes solen ser plataformes amb un “bot-as-a-service” on es pot construir, adaptar i desplegar un servei al cloud. Algunes d’aquestes opcions inclouen BotEngine, Microsoft Bot Framework, DialogFlow o IBM Cloud Watson. Totes elles proveeixen un set de funcionalitats, integracions i diferents nivells de usabilitat.

A més, els ChatBots sovint requereixen un backend amb una base de dades, on tornarem a afrontar una tria. Les opcions van des de les bases de dades estàndard SQL, amb una estructura poc compatible amb llenguatge natural, o d’altres més accessibles com les bases de dades NoSQL. També és pot triar la alternativa de tenir un backend basat en Graphs, com nosaltres. Les quèries en una base de dades basades en graph sovint s’assemblen a les connexions presents al llenguatge natural, el que les fa ideals per construir ChatBots.

Pel que fa a la eina de processat de llenguatge natural, tot i que alguns dels autors del article havien treballat anteriorment amb OpenNLP i altres eines pel processat, Dialogflow les guanya – per molt – tant en termes de precisió com d’usabilitat. Dialogflow proveeix eines per mantenir els contextos i et permet afegir respostes personalitzades quan les quèries fallen.

Aquestes respostes han de tenir un fonament amb dades reals, pel que caldrà crear un servidor entremig que gestioni el missatge de DialogFlow, el peticioni a la BBDD, i contesti a DialogFlow novament. Es vol tenir el màxim coneixement possible de les pel·lícules, pel que emprarem les dades d’IMDb, la font més popular del món de contingut de pel·lícules, televisió i celebritats

# Índex

[Introducció 1](#_Toc105546207)

[Índex 2](#_Toc105546208)

[Antecedents 3](#_Toc105546209)

[Presentació de la Proposta 3](#_Toc105546210)

[DialogFlow 3](#_Toc105546211)

[Base de Dades 4](#_Toc105546212)

[Servidor 4](#_Toc105546213)

[Experimentació / resultats 5](#_Toc105546214)

[Discussió 5](#_Toc105546215)

[Conclusions i Línies de Futur 5](#_Toc105546216)

[Agraïments 5](#_Toc105546217)

[Referències 6](#_Toc105546218)

[Informació Addicional 6](#_Toc105546219)

# Antecedents

# Presentació de la Proposta

Pel que fa a la nostre proposta, dividirem l’explicació en 3 apartats, donat que tenen una lògica ben diferenciada i una raó de ser també molt diferent. Aquests tres apartats seran DialogFlow, Servidor i Base de Dades. Part de la complicació del chatbot deriva de com gestionem aquests 3 aspectes i el procés que hem seguit per acabar tenint-ho com a la Figura 1.

Podem veure com aquí s’interacciona amb Telegram, però aquest parlarà amb DialogFlow qui també peticionarà al seu webhook, i aquest parlarà a través de GraphQL amb una base de dades Neo4J.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***Figura 1: Arquitectura de Theatro***

## DialogFlow

Per la creació del DialogFlow s’han seguit uns passos molt senzills, primerament s’ha creat el agent de pel·lícules a la plataforma:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

***Figura 2: Movie Agent***

Aquest és essencialment el vostre ChatBot. Podem configurar un agent preexistent o crear-ne un a través de la consola. Com hem fet nosaltres. L’hem anomenat MovieAgent.

Dins d’aquest trobem dos caps importants. Aquests seran els que ens permetran veritablement interactuar amb el chatbot e interpretar amb llenguatge natural el que ens estan dient.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***Figura 3: Intents & Entities***

En primer lloc tenim els Intents. Aquests capten la intención d’un usuari que interactua amb el nostre agent. Qualsevol cosa que l’usuari escrigui/parli amb l’Agent, s’assigna a una intenció. Per defecte en tenim dues de “Fallback”. A més d’aquests n’hem creat 5 més:

Aplicación

Descripción generada automáticamente con confianza media

***Figura 4: Els nostres Intents***

Es pot veure com s’han creat intents per cada una de les funcionalitats del chatbot. Tenim un intent per el processat d’una recomanació, quan ens demani recomanar-li alguna pel·lícula. Tenim un intent dedicat a la creació del bot, a ajuda, a informació i a agraïments.

Cada un d’aquests ha estat entrenat amb un munt d’exemples d’input per part de l’usuari, i d’entities.

Què son aquestes Entities?. Si pensem en el nostre MovieAgent, quan l’usuari diu: “Dona’m informació sobre Jobs”, això hauria de dir a l’agent que necessito informació sobre la pel·lícula: Jobs. Però com sap això, com és capaç de analitzar amb aquest detall la fase? Bé, per cada expressió d’usuari assignada a una intenció, el nostre agent necessita informació. Aquesta la obté amb ajuda de les Entitats. Cada expressio d’usuari d’una intenció pot tenir una entitat que defineix la cosa, de la qual l’usuari necessita informació.

Imagen que contiene Tabla

Descripción generada automáticamente

***Figura 5: Les nostres Entities***

Aquestes son, categoria de la pel·lícula, son els tags pels quals pot identificar-se una pel·lícula, ja sigui gènere o simplement adjectius que la descriuen. Content-type, si volem pel·lícules, series, curts, etc. Hem declar una de movie, per que sigui capaç d’entendre noms de pel·lícules i series, un de roles per tots els possibles rols d’una producció, per que es pugui cercar amb aquesta restricció, i finalment una entity dinàmica de cerca, que combina les anteriors entre sí.

És important mencionar com, a més de les nostres entities, DialogFlow té les seves entities pròpies, com ara la detecció de números o noms propis.

Havent definit tot això, el que va caldre és entrenar amb molts exemples a aquesta IA, per que pogués saber quin tipus de missatges esperar-se. Com no és d’estranyar, Google és molt potent, i només cal posar un seguit d’exemples i ja és capaç d’entendre’n de nous amb els que no se l’hagi entrenat.

A continuació, calia vincular la pròpia interfície gràfica. El propi DialogFlow ofereix un seguit de integracions directes amb un munt d’eines com ara Slack, Line, Skype o Telegram. En el nostre cas, vam voler triar Telegram donat que creiem que és la més potent de totes elles, i ja l’havíem emprat anteriorment per l’altre Chatbot i vam creure adient fer-lo servir per aquest.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

***Figura 6: Configuració Telegram***

Per últim, vam afegir un extra que ofereix DialogFlow que és la possibilitat de tenir converses trivials amb el Chatbot. Preguntar-li si està feliç, dir-li que estàs trist, o que no has entès la resposta/pregunta realitzada. És el que DialogFlow anomena “Small Talk”, una funcionalitat que permet que de manera intel·ligent, i novament amb una IA de Google ja entrenada Theatro se senti més humà, i no tingui moments on no sàpiga que fer.

## Base de Dades

Havent definit com es comunicarà

BBDD ha seguit aquests canvis:

1. API
2. PostgreSQL
3. PotsgreSQL + Agens
4. PostgreSQL + Hasura (GraphQL)
5. Neo4J
6. Neo4J + GraphQL

## Servidor

El servidor és una part molt notable del nostre Chatbot, és qui rep les dades i les peticiona a la base de dades

# Experimentació / resultats

Joc de proves / Hardware / Software

Com es faran els experiments / Mètriques

Experiments o Casos pràctics

Resultats

# Discussió

És una valoració més qualitativa

# Conclusions i Línies de Futur

Fins ara cap dels membres del grup havia treballat mai amb la enorme majoria de tecnologies emprades per aquest projecte. El plantejament inicial, a la versió basada amb regles plantejada anteriorment a l’assignatura de Sistemes Basats en el Coneixement, vam poder fer una primera aproximació al que seria el processat del llenguatge natural e intentar que aquest resulti en una resposta per part del servidor el més clara possible I amb la informació més detallada possible. No teníem base de dades, aquestes venien d’una api externa, i el processat del llenguatge era molt rudimentari i limitat.

Theatro és diferent. Amb Theatro hem pogut posar en pràctica un ventall d’eines i coneixements molt gran. Ens hi hem barallat per decidir quina tecnologia fèiem servir de “frontend” per poder fer el processat del llenguatge natural. Inicialment vam voler implementar la nostre pròpia IA, el que ens va portar a fer una bona recerca en aquell aspecte, aquesta va derivar en el ús d’eines com DialogFlow per deixar aquell aspecte del Chatbot a un servei de tercers molt més potent del que haguéssim pogut fer nosaltres. Això sí, la nostre ambició seguia sense estar satisfeta.

És per això que vam voler fer que les dades amb les que treballés el DialogFlow fossin el més robustes i ràpides possibles. Per fer-ho, tal i com hem detallat a la nostre proposta, vam provar un ventall enorme de tecnologies que ens vam permetre aprendre un munt, portant-nos finalment a emprar Neo4J com a eina per la base de dades i GraphQL com a “Middleware” per realitzar les quèries.

No és el primer cop que treballem amb bases de dades basades amb graphs però sí el primer que ho fem amb tant de detall, i amb una base de dades tan gran.

Ha estat un procés d’autoaprenentatge enorme i de descobriment d’habilitats i eines que no sabíem que teníem. Però havent fet menció a la nostre ambició, un podria creure que aquesta ha estat satisfeta, però no és el cas.

Tenim moltes idees per línies de futur, que no hem pogut implementar per manca de temps però que estaven a la versió inicial de l’entrega. Aquestes passen per una millora estètica dels missatges presentats, per obrir més el ventall de quèries que es poden realitzar, donat que tot i ser molt ampli ara mateix hi ha dades que no estem preparats a donar, com ara el link del tràiler d’una pel·lícula, o preguntes niades o similars. Per la comprensió d’aquestes també ens hagués agradat afegir d’una forma més robusta la concatenació de condicions amb condicionals com “or” i “and” donat que ara mateix esta fet d’una manera més rudimentària a la part de servidor, però tenint en compte que podem emprar DialogFlow per tasques com aquesta, estem convençuts que podríem acabar obtenint resultats molt superiors als actuals.

# Agraïments

Els autors els agradaria esmenar i agrair el suport ofert per la professora Elisabet Golobardes Ribé de La Salle. Els agradaria també agrair l’ajut dels becaris de l’assignatura de Sistemes Basats en el Coneixement, Martí Ejarque i Alejandro Moñux. A la institució de La Salle Campus Barcelona per proporcionar els recursos, eines i coneixements necessaris per tenir un punt de partida sòlid. També volen esmenar la enorme importància de la disponibilitat gratuïta d’eines com DialogFlow oferta per Google i l’eina Heroku oferta per SalesForce.

# Referències

# Informació Addicional

Dibujo en blanco y negro

Descripción generada automáticamente con confianza mediaThis work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/