Kibo: un xatbot diferent

Ariel Andreas Daniele, Daniel Mezo Ginebreda i Clara Valls Llovera

Departament d’Enginyeria

La Salle, Universitat Ramon Llull

Barcelona

# Resum

En aquest article es presentarà el xatbot implementat durant la realització del projecte de l’assignatura de Sistemes Basats en el Coneixement. Aquest xatbot consisteix en un joc interactiu entre l’usuari i l’aplicació, narrat pel propi xatbot, que permetrà al jugador construir la seva pròpia història a partir de les eleccions que faci.

Tenint en compte que el xatbot serà l’intermediari entre l’usuari i l’aplicació, l’objectiu principal que es tractarà serà el de crear un sistema capaç d’interpretar correctament els missatges de l’usuari, per així donar-hi una resposta adequada que li permeti seguir el transcurs de la història.

Cal tenir en compte que el sistema està desenvolupat de manera que queda integrat a l’aplicació de Telegram i, per tant, només funcionarà en aquest entorn.

# Paraules clau

Kibo, xatbot conversacional, història interactiva, machine learning,...

# 1. Introducció

El món dels xatbots ha evolucionat molt i, com era d’esperar també tindria la seva incidència en els videojocs. Si ens fixem ens els propis videojocs en tenim de tot tipus, però sempre hi ha una necessitat de satisfacció del client. L’objectiu és divertir, i si no és el cas, almenys entretenir. El que volem amb el nostre xatbot és que quan l’usuari acabi la història tingui la sensació que el temps emprat hagi estat profitós. El temps, un tret que a tothom li importa més del que sembla. Una realitat amb la que possiblement tothom estigui d’acord és que no ens agrada perdre el temps. Volem que el temps invertit tingui alguna recompensa o satisfacció. Per això ens hem bolcat a implementar un xatbot que a l’usuari li suposi un esforç temporal, però que al mateix temps no interpreti aquest temps que li està dedicant com a desaprofitat, sinó tot el contrari, ha de sortir més enriquit que abans. L’única feina real d’en Kibo és entretenir a l’usuari i mantenir una conversació animada i interactiva, on les respostes de l’usuari tinguin un paper fonamental en el desenvolupament de la conversa, tal i com passaria amb dos humans. També es tracta que cada persona tingui la seva relació personal i avanci de manera diferent per la història. Podem tenir un amic en comú, però les històries i els moments amb ell no es poden replicar per cap altra relació. Doncs el mateix intenta Kibo, qui vol ser amic de tots però al mateix temps ajudar a que cadascú agafi el camí que prefereix, sense pressió, acceptant i acompanyant-lo al llarg de la seva pròpia aventura.

# 2. Estat de l’art

En el mercat hi ha milers de xatbots i cada dia en sorgeixen de nous. Però, en el nostre cas concret no n'hi ha cap, almenys que estigui molt desenvolupat, que abordi la mateixa manera la necessitat d’entretenir. Existeixen assistents virtuals per videojocs, com el cas de Halo [1] o Pokémon Go [2], però precisament perquè la dificultat i la gran quantitat de dades que gestionen aquests entorns requereixen d’alguna eina intel·ligent. També podem buscar una certa equivalència amb la pel·lícula ‘Black Mirror: Bandersnatch’ [3], la qual és considerada una de les primeres pel·lícules interactives, on l’espectador pot decidir com continuarà la història segons les decisions que faci per pantalla.

Malgrat la gran varietat que existeix, els xatbots actuals se centren poc en el domini que hem escollit, com són les històries interactives. Per aquest motiu, hem volgut implementar una conversa que, tot i ser bàsicament missatges de text, a l’usuari li inspiri a descobrir quina serà la següent decisió a prendre i les conseqüències que podria tenir. Aquesta sensació de poder, on el desenvolupament està estrictament relacionat amb les decisions, fa que l’usuari es senti protagonista, inclòs en una conversació on l’aportació personal és molt menor a la de l’interlocutor, però només en quantitat, ja que només una paraula pot canviar l’enfoc bruscament.

# 3. Base de coneixement

Un cop detectat el problema a resoldre i s’ha especificat el domini en el que es troba, caldrà definir com serà el coneixement que es proporcionarà al xatbot.

En el cas que ens ocupa, les úniques dades que obté el sistema de l’exterior són les proporcionades per un fitxer en format JSON. Aquest, conté tota la informació necessària pel desenvolupament del joc i, alhora, permet ampliar-ne fàcilment el contingut sense necessitat de modificar cap part del codi.

Així doncs, el coneixement que té el xatbot és relacional simple, ja que no requereix de cap procediment lògic o d’herència per trobar les respostes que busca. L’únic mecanisme d’inferència que es realitzarà serà consultar la informació proporcionada pel fitxer JSON, que seria l’equivalent a una taula relacional simple.

La representació que es farà d’aquest coneixement serà completa, monòtona i consistent. És completa perquè es considera que no faltaran mai dades per resoldre el problema: allò que no es trobi en el fitxer de dades es considerarà que no existeix, fent així una assumpció del món tancat. És monòtona perquè es pot afegir fàcilment coneixement nou, i l’anterior no quedarà invalidat en cap cas, és a dir, es poden afegir parts noves de la història que relata el joc sense que les preexistents es vegin modificades. És consistent perquè no hi haurà dades errònies ni contradictòries: els nodes que componen informació del fitxer de dades són completament independent entre ells, de manera que mai es podran contradir.

Així doncs, es podria considerar que la Base del Coneixement que es construeix contindrà un nombre N de casos concrets, i sempre trobarà una solució al problema que se li plantegi, sempre i quan es trobi dins d’aquests N casos. En cas contrari, considerarà que el problema que se li planteja no el pot resoldre perquè l’usuari no ha introduït correctament la informació.

# 4. Resolució del problema

Per la implementació d’aquest projecte es requereixen diversos algorismes.

Com que es tracta d’un xatbot conversacional, és necessari tractar els missatges enviats per l’usuari per donar-los una resposta adequada. El xatbot Kibo és un joc interactiu que permet a l’usuari escollir el transcurs de la seva pròpia història.

Per la qual cosa, en funció de a quina punt de la història es trobi, l’usuari haurà de respondre entre unes opcions concretes per seguir per un camí o per un altre.

Un dels problemes a resoldre és la recepció dels missatges de l’usuari. L’algorisme utilitzat per aconseguir-ho es podria simplificar amb la següent implementació del programa principal:

funcio main()

mostraIntroduccio()

mentre (1)

resposta = recullMissatge()

respon(resposta)

fimentre

fifuncio

Com es pot veure, el programa principal mostra primer una introducció per situar a l’usuari a la història. Aquest explica el funcionament del xatbot de forma simple, presentant així al personatge anomenat Kibo, que és qui explica la història.

El principal problema que apareixerà serà interpretar correctament el missatge rebut de forma que el xatbot l’entengui. Per fer-ho caldrà fer un tractament del llenguatge escrit. La funció que farà aquest tractament es podria simplificar amb el següent algorisme:

funcio respon(cadena resposta)

opcio = tractaResposta(resposta)

enviaResposta(opció)

fifuncio

Aquesta funció serà l’encarregada de fer el tractament del llenguatge i obtenir el següent camí pel qual seguir l’execució. El valor d’aquest node permetrà enviar la resposta adequada a l’usuari per poder continuar amb la història.

Com s’ha dit anteriorment, el principal problema que es detecta és el de la interpretació de la resposta donada per l’usuari. La funció encarregada de resoldre’l té un funcionament simple.

Com que en tots els casos, la resposta de l’usuari està limitada a dues possibles opcions que se li ofereixen, el sistema esperarà rebre una cadena que contingui unes paraules concretes. D’aquesta manera, cadascun dels nodes que formen l’arbre de la història, contenen el missatge amb la pregunta i les paraules que s’espera rebre com a resposta de l’usuari. Per cada opció que se li ofereix a l’usuari hi ha assignades dues paraules, que s’espera que, almenys una de les dues es trobi a la cadena que forma la resposta de l’usuari. És a dir, si, per exemple, l’usuari es troba en un punt de la història on pot escollir si vol endinsar-se a la selva o explorar la costa, la resposta que s’espera pot contenir les paraules “selva” o “endinsar” per la primera opció, i “costa” o “explorar” per la segona.

Cal destacar que, en el cas dels verbs, s’ha comparat únicament amb l’arrel d’aquest, acceptant així qualsevol terminació del verb per permetre respostes en qualsevol persona i temps verbal.

Per tant, l’algorisme que s’encarrega de fer el tractament del llenguatge seria el següent:

funcio tracta Resposta (cadena resposta)

si (resposta conté paraula1)

retorna opcio1

si (resposta conté paraula2)

retorna opcio1

si (resposta conté paraula3)

retorna opcio2

si (resposta conté paraula4)

retorna opcio2

sino

retorna error

fifuncio

Com es pot observar, aquesta funció comprova si la resposta rebuda conté alguna de les paraules clau que s’esperen obtenir i, en funció d’això retorna l’opció desitjada, o error en cas de desconèixer la resposta.

La combinació dels algorismes vistos permet implementar el sistema que resol el problema plantejat.

# 5. Sistema

Un sistema basat en el coneixement és un programa informàtic que requereix d’una base de coneixements i un motor d’inferència per resoldre problemes.

En aquest projecte, com s’ha especificat anteriorment, la Base de Coneixements serà un sistema tancat amb un nombre concret de casos. Aquesta obtindrà la informació d’un fitxer de dades, el qual contindrà la història i els possibles camins que pot escollir l’usuari.

Per representar la història i els possibles camins que pot prendre l’usuari s’ha utilitzat un arbre, on cadascun dels seus nodes seran les parts de la història a les que correspon aquell camí i, les fulles, seran els possibles finals. En funció de les eleccions de l’usuari, es recorreran uns nodes o uns altres.

Per tant, l’únic raonament que farà el motor d’inferència serà buscar al fitxer de dades la informació que correspongui al node de l’arbre en el que es trobi l’usuari.

Tenint això en compte, es podria dir que el xatbot s’ha dissenyat al voltant d’aquest fitxer de dades, on es troba tot el coneixement del xatbot. Aquest conté una llista de nodes, corresponents als nodes de l’arbre, on de cadascun n’especifica la següent informació:

* **id**: identificador únic del node.
* **text**: missatge relatiu a aquell node que s’enviarà a l’usuari si escull aquell camí.
* **següent**: llista de dos valors que conté els fills, esquerra i dret en aquest ordre, del node actual. Si l’usuari escollís la primera opció, el següent node a tractar seria l’esquerra i si escollís la segona seria el dret.
* **key\_words**: aquest camp conté les paraules clau que s’utilitzaran per fer el tractament del llenguatge. Consisteix en una llista de quatre valors: els dos primers són les paraules o parts de paraula que s’esperen rebre si la resposta correspon a la primera opció, i els dos últims seran les relatives a la segona opció.
* **end**: aquest campcorrespon a un booleà que indica si el node en qüestió és un node fulla i, per tant, correspon a un final de la història.

Serà necessari conèixer el node actual que s’està tractant per poder obtenir-ne la seva informació i fer el tractament adequat de cada missatge. Això es farà amb una variable global que emmagatzemi el valor de l’identificador del node. Cada cop que es rebi un missatge de l’usuari, caldrà consultar aquest valor per saber en quin punt de la història es troba i prosseguir a tractar el missatge adequadament. Un cop el sistema reconeix quina de les dues opcions ha escollit l’usuari, actualitzarà el valor del node actual i enviarà el missatge que correspon al següent node a tractar.

El xatbot s’encarrega únicament de fer d’intermediari entre l’usuari i el sistema. És qui rep i envia els missatges d’un a l’altre per poder procedir amb la història.

Es va decidir utilitzar una API existent preparada pel desenvolupament de xatbots que, a més permetia integrar aquest a l’aplicació de Telegram. Aquesta API s’encarrega únicament de recollir les respostes de l’usuari i enviar-li els missatges corresponents.

Les funcions per recollir el missatge enviat per l’usuari i enviar-hi una resposta es troben a una funció pròpia de la API de Telegram. És necessari crear una classe, serà l’anomenada KiboBot.java, que estengui de la classe pròpia de l’API anomenada *TelegramLongPollingBot*. En ella hi ha les funcions que permetran la comunicació entre l’aplicació i l’usuari.

Tot i així, l’API esmentada no ofereix cap funció de tractament de llenguatge. Per fer-lo, s’hauran d’utilitzar els algorismes vistos a l’apartat anterior. Aquests permeten dissenyar el funcionament del xatbot perquè pugui rebre i enviar missatges a través de la funció de la API i, a més, pugui tractar les respostes correctament.

Així doncs, es podria resumir el funcionament global del xatbot amb el següent esquema, que mostra el recorregut que segueix el codi mentre s’executa:

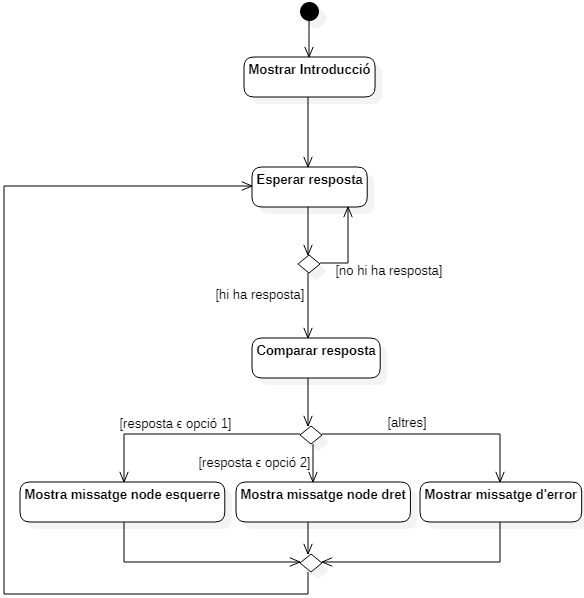


Figura : Diagrama d'activitats del codi del xatbot

# 6. Experimentació

La primera fase d’experimentació s’ha portat a terme durant els transcurs de la implementació del nostre xatbot. En aquesta, nosaltres mateixos hem anat fent petits jocs de proves per poder validar el funcionament del joc. Tot i així, aquest no és un sistema de proves eficient, ja que l’usuari final seran les persones externes que sense cap coneixement sobre el xatbot en sí. Això pot provocar que l’usuari final introdueixi qualsevol paraula com a resposta que nosaltres, com a desenvolupadors, no haguem tingut en compte. Així que, una vegada es finalitza la primera versió del xatbot, es passaria a una segona fase de proves.

Durant aquesta segona fase, el que ens interessaria seria fer un petit test amb unes 50 persones de totes les edats. Aquestes haurien de provar el xatbot i realitzar una petita enquesta on es podrien obtenir diferents resultats: si els hi ha agradat o no, si tenen temps de llegir tots els missatges o es perden alguna mena d’informació o quin percentatge de les respostes que donen són esteses pel xatbot. En funció dels resultats obtinguts ens enfocaríem a millorar els diferents punts negatius obtinguts a través de les enquestes. Llavors començaria una tercera fase d’experimentació.

En aquesta tercera fase, ens enfocaríem a realitzar les millores necessàries per solucionar els punts negatius observats durant l’anterior fase de proves. Al finalitzar les millores, caldria tornar a la fase segona per veure com ha millorat el xatbot i, detectar també nous errors o problemes que no s’han resolt correctament. Si tot surt bé ja estaria finalitzada la primera implementació per presentar.

# 7. Discussió

El primer obstacle que vam afrontar va ser decidir amb quina eina realitzar el nostre xatbot. Vam començar investigant les eines existents que ens permetien desenvolupar el nostre xatbot de la millor forma possible. Com a desenvolupadors ens sentim molt còmodes treballant amb Java, així que vam buscar eines que ho permetessin. Finalment vam trobar una API que ens permetia integrar el nostre xatbot a Telegram i, a més, es podia desenvolupar en Java.

Un cop decidit amb quina eina anàvem a desenvolupar el nostre xat bot, ja ens podíem posar mans a l’obra.

Vam escollir un tema que fos força interesant per tots els membres del grup i, a més, ens motivés per desenvolupar el nostre xatbot de la millor forma possible. Al tractar-se d’un joc, el nostre xatbot es converteix en una manera de passar el temps, divertit i que permet posar-se a la pell del protagonista per viure una experiència única.

Creiem que el nostre xatbot és molt escalable, ja que té un gran marge de millora i creixement. De moment nomes tenim una història amb diversos finals, però es podrien afegir fàcilment noves parts de la història o fins i tot crear-ne de noves. Això ens faria arribar a un altre nivell d’experiència per l’usuari.

En el nostre cas el tractament del llenguatge natural que hem hagut de realitzar ha sigut molt mínim, ja que vam escollir limitar les opcions de l’usuari per facilitar la interpretació de les seves respostes. Això redueix una mica la llibertat de l’usuari a l’hora de prendre una decisió, ja que només pot escollir entre dues opcions.

A més, el xatbot treballa en un entorn tancat, és a dir, no necessita més informació de l’exterior de la que té a l’inici de la seva execució, fet que ha facilitat el seu desenvolupament.

# 8. Conclusions

Com a conclusió final, podem dir que ens ha agradat molt realitzar aquest projecte, ja que és una eina cada cop més utilitzada, però amb la que mai ens hi havíem afrontat com a desenvolupadors. A més, al poder escollir el tema que havia de tractar, ens ha permès portar-ho al nostre terreny i gaudir-ho encara més.

Vam escollir un tema que ens resulta força interesant i, amb el qual, podíem tenir un gran marge de creixement i millora, ja que els videojocs poden tenir una gran varietat de material a oferir. En nostre objectiu era crear una nova forma de passar el temps, però a l’hora de divertir-te i gaudir d’una aventura amb la qual passaran moltes coses emocionants durant el seu transcurs.

La realització d’aquest treball, ens ha servit per entendre com funciona un xatbot internament i com s’ha d’intentar adaptar a les diferents situacions. Tot i que no hem aconseguit tenir un número elevat d’usuaris amb els que testejar el xatbot, hem demanat a familiars i amics que el provessin. Això ens ha permès obtenir un feedback per anar fent petites modificacions per millorar-lo i adaptar-lo al que els usuaris esperaven com, per exemple, si alguna part no s’entenia, o si esperaven trobar més elements visuals, per exemple.

Com a línies de futur, ens agradaria poder fer diferents histories i diferents personatges amb els que realitzar les aventures, per així poder gaudir l’experiència des de diferents punts de vista. També ens agradaria molt poder introduir sons i sorolls per tal de crear una experiència molt més real. A més, ens agradaria poder realitzar els nostres propis dissenys dels personatges i de les imatges del joc per donar-li un toc més visual i personal al nostre xatbot. Per últim, creiem que podríem arribar a millorar el funcionament del xatbot per tal que acceptés respostes molt més obertes dels usuaris, i crear així experiències més personals i úniques per cada jugador. Això implicaria millorar el tractament del llenguatge de les respostes de l’usuari, per així poder donar llibertat total a l’usuari durant el transcurs de la història.

# Referències

[1] Xatbot Halo. Cortana. Microsoft Bing Satori. 2014

[2] Pokémon Go Chat Bot. Chat-bot to help Pokémon trainers become Pokémon Masters. 2016

[3] Xataka. ‘Black Mirror: Bandersnatch’ es la primera película interactiva de Netflix: así funciona. 2018