



Universitat Politècnica de Catalunya

FACULTAT D'INFORMÀTICA DE BARCELONA

DISSENY DE SISTEMA DE DIÀLEG
BASAT EN REGLES

Tractament de la Veu i el Diàleg

Autors:

Zhihao Chen; Zhiqian Zhou

2 d'octubre de 2024

Índex

Introducció	2
Exercicis i els seus resultats.	2
2.1 Fitxer JSON	2
2.1.1 Exercici 1	2
2.2 Sistema de diàleg senzill	3
2.2.1 Exercici 2	3
2.2.2 Exercici 3	4
2.2.3 Exercici 4	4
2.3 Millorem el sistema de diàleg	5
2.3.1 Exercici 5	5
2.3.2 Exercici 6	6
2.3.3 Exercici 7	6
Conservation flow	8
Conclusió	10

1 Introducció

En l'era de la digitalització, la interacció entre humans i màquines mitjançant el llenguatge natural s'ha convertit en un recurs fonamental per oferir serveis personalitzats i accessibles.

Aquest treball se centra en el disseny d'un xatbot basat en regles, dissenyat per recomanar pisos disponibles segons les necessitats de l'usuari. El sistema seguirà un esquema de preguntes predefinides i farà ús d'una base de dades actualitzada de propietats.

El xatbot s'implementarà utilitzant Python, un llenguatge de programació versàtil que facilita el desenvolupament de xatbots gràcies a la seva àmplia comunitat i la disponibilitat de llibreries especialitzades, com NLTK per al processament del llenguatge natural. A través d'un diàleg estructurat, el sistema formularà preguntes a l'usuari sobre aspectes com la ubicació desitjada, el tipus d'immoble (lloguer o compra), el pressupost, etc. Amb les respostes obtingudes, el xatbot consultarà la base de dades i oferirà una llista de recomanacions ajustades a les preferències especificades.

El projecte es basa en un sistema de diàleg controlat per regles, on cada interacció està estructurada segons patrons predefinits per garantir que l'usuari rebi la informació més rellevant. Aquest enfocament permet un control precís del flux de la conversa i la lògica de selecció de pisos, assegurant respostes adequades basades en criteris específics.

Progressivament, i mitjançant una sèrie d'exercicis, s'aniran incorporant noves funcionalitats per fer el sistema més complet i eficient.

2 Exercicis i els seus resultats

2.1 Fitxer JSON

2.1.1 Exercici 1

En el primer exercici, se'ns demana afegir una nova instància (casa) a la base de dades disponible. Si revisem la base de dades proporcionada, observarem que està estructurada en un fitxer JSON amb quatre claus principals:

- **start message:** Un missatge (cadena de caràcters) de benvinguda que es mostra a l'usuari en iniciar la interacció.
- **end message:** Un missatge (cadena de caràcters) que acomiada l'usuari quan finalitza o abandona el diàleg.
- **questions:** Un conjunt de preguntes, cadascuna representada com un diccionari que conté informació com el tipus de pregunta (numèrica o de selecció múltiple), el prompte i una clau identificadora de la pregunta.
- **houses:** Una llista de propietats disponibles. Cada casa té els seus atributs, com ara el número d'identificació, el tipus d'immoble, el nombre de dormitoris i lavabos, el preu, la superfície, la

ubicació, i si disposa d'ascensor, terrassa o ús comercial.

Per tant, per incorporar noves instàncies a la base de dades, només cal afegir-les a la llista de cases, juntament amb les seves característiques pròpies, tal com ens indica l'exercici. No obstant això, és important tenir en compte que el número d'identificació (ID) de cada instància ha de ser únic, de manera que la nova casa afegida ha de tenir un ID diferent dels ja existents.

```
1 new = {"id": 26, "type":"sale", "bedrooms": "1", "bathrooms": "1", "price": "100k", "
    square_meters": "50", "floor":"1", "elevator":"No", "commercial_use":"No", "
    terrace": "No", "location": "Santa Coloma de Gramenet"}
2
3 data['houses'].append(new)
```

La sortida al terminal després d'imprimir les dades és:

```
House ID: 26
Bedrooms: 1
Bathrooms: 1
Price: 100k
Square Meters: 50
Location: Santa Coloma de Gramenet
```

2.2 Sistema de diàleg senzill

En aquesta secció, s'han definit diverses funcions auxiliars per processar les dades i les respostes, que permeten crear un sistema senzill capaç de mostrar preguntes predefinides i emmagatzemar la informació rellevant de les respostes de l'usuari.

2.2.1 Exercici 2

En l'exercici 2, se'ns demana mostrar la frase de benvinguda que es troba al fitxer JSON a l'usuari quan s'inicialitza el diàleg. Per facilitar la definició de les funcionalitats del sistema i l'organització del codi, crearem una classe anomenada 'Chatbot', que actuarà com a model de gestió del diàleg. Copiem els codis de processament de preguntes i els incorporem dins de la classe. A més, definirem un mètode anomenat 'init', que s'encarregarà d'inicialitzar el diàleg, cridar als mètodes de processament de preguntes per mostrar les preguntes predefinides i guardar les respostes de l'usuari en una variable d'instància ("user_preference").

Per tant, per mostrar la frase de benvinguda, només cal imprimir-la en inicialitzar el diàleg, és a dir, quan l'usuari crida al mètode 'init' de la classe:

```
1 class Chatbot():
2     def __init__(self, data):
3         ...
4
5     def init(self):
6         ...
7         print(self.data['start_message'])
8         ...
```

```

9
10     def ...
11
12 user_1 = Chatbot(data)
13 user_1.init()

```

La sortida, per tant, és l'esperada:

```

Welcome to the House Buying Assistant!
How many bedrooms do you need?
...

```

2.2.2 Exercici 3

En aquest exercici, s'ha afegit la funcionalitat que permet a l'usuari sortir del diàleg en qualsevol moment de la conversa. Per aconseguir-ho, hem incorporat una condició tant en el processament de preguntes numèriques com en el d'opcions múltiples. Després d'obtenir la resposta de l'usuari, es comprova la següent condició:

```

1 if answer.lower() == 'quit':
2     print(self.data['end_message'])
3     sys.exit()

```

Aquesta implementació permet que el sistema mostri el missatge d'acomiadament que es troba al fitxer JSON en qualsevol moment del diàleg, responent “quit” a qualsevol pregunta del xatbot. A més, no cal preocupar-se per si la paraula conté majúscules o minúscules, ja que es fa servir el mètode “lower()” per realitzar la comparació de manera insensible a la capitalització. En provar-ho, la sortida és l'esperada:

```

Welcome to the House Buying Assistant!
How many bedrooms do you need?
Enter your choice (1 - 5 bedrooms): Quit
Thank you for using the House Buying Assistant. Goodbye!
...

```

2.2.3 Exercici 4

Fins ara, el nostre xatbot encara estava incomplet en el sentit que no feia recomanacions de cases, tal com s'havia plantejat inicialment. En aquest exercici, l'objectiu és que, a partir de les preferències de l'usuari emmagatzemades en la variable “user_preferences”, el xatbot sigui capaç de filtrar i obtenir les cases que s'ajustin a aquestes necessitats per, posteriorment, recomanar-les a l'usuari.

Per aconseguir-ho, hem dissenyat la funció “find_suitable_houses”, que itera sobre totes les cases de la base de dades i guarda aquelles que compleixen amb les preferències de l'usuari. Aquest filtratge es realitza mitjançant un diccionari anomenat “filters”, on les claus corresponen a les “answer_key” i els valors són funcions que comparen les preferències de l'usuari amb les característiques de cada casa. Perquè una casa sigui recomanada, ha de passar totes les funcions de comparació del filtre.

En provar el sistema, si executem “user_preferences”, obtenim:

```
{'bedrooms': '1',  
  'bathrooms': '1',  
  'price': '120k',  
  'square_meters': '80',  
  'location': 'Barcelona'}
```

Després d'executar la funció “print_suitable_houses”, obtenim els següents resultats:

```
Based on your preferences, the most suitable houses are:  
House ID: 22  
Type: rent  
Bedrooms: 4  
Bathrooms: 2  
Price: 3000 euros  
Square Meters: 100 m^2  
Location: Barcelona
```

El sistema realitza correctament la recomanació, tal com es pot comprovar manualment en la base de dades.

Cal destacar que la majoria de les comparacions no es basen simplement en la igualtat. Per a les preguntes numèriques, sovint es fan comparacions de superioritat o inferioritat, ja que les preferències de l'usuari s'entenen com un límit que defineix les restriccions.

2.3 Millorem el sistema de diàleg

2.3.1 Exercici 5

En els exercicis anteriors, hem observat que sempre era necessari triar una opció específica. El propòsit d'aquest exercici és permetre que l'usuari pugui introduir l'opció “any”, oferint així una recomanació més àmplia en cas que no es vulgui limitar a un tipus de casa concret.

Per aconseguir-ho, hem afegit a la inicialització de la classe “Chatbot” una variable anomenada “self.any”, que és una llista on es guarden els noms dels “answer_key” per als quals l'usuari ha triat l'opció “any”. A continuació, en les funcions (o mètodes) que gestionen les preguntes numèriques i d'opcions múltiples, hem inclòs una nova condició: si l'usuari introdueix “any” (sense importar si conté majúscules), l’“answer_key” corresponent es guarda a la llista “self.any”.

La llista “self.any” és útil durant el filtratge de cases dins de la funció “find_suitable_houses”. Anteriorment, cada casa que es recomanava havia de complir tots els requisits del filtre. Ara, amb aquesta extensió, si una propietat correspon a algun dels criteris a “self.any”, no serà necessari comprovar aquest criteri en el filtre, ja que automàticament es considera que compleix les necessitats de l'usuari.

Per validar aquesta funcionalitat, com a simplificació, responem “any” a totes les preguntes. En aquest cas, el sistema simplement passa a la següent pregunta cada vegada que es respon “any”, i a la recomanació final es mostren totes les cases disponibles, ja que no s'han imposat restriccions.

```
...  
Based on your preferences, the most suitable houses are:
```

```
House ID: 1
...

House ID: 2
...

...

House ID: 25
...

House ID: 26
...

Thank you for using the House Buying Assistant. Goodbye!
```

2.3.2 Exercici 6

A la base de dades, els aspectes de les cases no estan limitats per les preguntes predefinides. Per tant, en aquest exercici s'ampliarà la sèrie de preguntes per tal que el "Chatbot" pugui oferir recomanacions més precises basades en preferències més concretes.

El sistema implementat anteriorment llegeix la base de dades per imprimir les preguntes, de manera que una forma d'ampliar-les és afegir noves preguntes directament al fitxer JSON. Editem el fitxer JSON tenint en compte que les noves preguntes estiguin correctament formatades i continguin tota la informació necessària per garantir el correcte funcionament del xatbot.

A continuació, cal modificar la implementació del xatbot, ja que hi ha un aspecte important a considerar que diferencia de les preguntes anteriors. Es tracta de la pregunta sobre el tipus d'habitatge que l'usuari prefereix, amb tres possibles opcions (lloguer, compra o "any"). Si l'usuari selecciona lloguer (o "any"), s'ha d'afegir una pregunta addicional relacionada amb el pressupost: Quin és el seu salari mensual? En aquest cas, a la recomanació de cases, el filtre exclourà aquelles propietats en què el preu del lloguer superi el 35% dels ingressos mensuals de l'usuari, sempre que el tipus d'habitatge sigui de lloguer.

Aquestes noves preguntes introdueixen condicions addicionals que les diferencien de les anteriors, ja que segons la resposta, el xatbot haurà de generar preguntes diferents a continuació. A més, en el procés de filtratge de cases, el sistema haurà de tenir en compte el tipus d'habitatge de cada propietat per, si l'usuari tria lloguer o "any", aplicar la condició addicional del pressupost en aquelles propietats destinades a lloguer.

2.3.3 Exercici 7

Gràcies a les implementacions anteriors, ja disposem d'un sistema de diàleg funcional. No obstant això, encara hi ha molts aspectes a millorar per tal d'oferir una millor experiència d'usuari. Aquest és precisament l'objectiu que es planteja en aquest exercici.

La millora del sistema es basa en la implementació d'un mètode iteratiu que segueix quatre processos clau:

- **Exemples:** En aquesta fase, es recullen exemples reals o simulats de converses amb usuaris per identificar quines parts del sistema poden fallar o resultar confuses.
- **Camins:** Un cop recopilats els exemples, es defineixen els diferents camins o fluxos de diàleg que l'usuari podria seguir durant la conversa.
- **Prototipatge:** En aquesta etapa, es crea un prototip del sistema millorat basat en els exemples i camins definits anteriorment.
- **Proves:** Finalment, es fan proves amb usuaris reals o simulacions per avaluar el rendiment del sistema i identificar possibles errors o millores addicionals.

El bucle es repeteix fins que el sistema ofereixi una experiència òptima.

- **Primera iteració:** Des de l'inici de la pràctica, vam detectar una de les limitacions del sistema: la seva capacitat d'interpretar correctament les respostes de l'usuari. Per a les preguntes numèriques, el sistema és capaç d'analitzar una oració i extreure'n els números, però sense comprendre el seu significat real. Per a les preguntes categòriques, la comprensió està encara més limitada, ja que només pot acceptar respostes directes. Per exemple, si per a la pregunta de localització l'usuari respon "I would like to live in Barcelona", el sistema rebutja la resposta fins que l'usuari respongui amb "Barcelona".

Una possible solució seria incorporar un model de comprensió del llenguatge natural, com ara el reconeixement d'entitats. Tanmateix, per simplicitat, hem decidit optar per una solució més senzilla: avisar l'usuari del format correcte que ha de tenir la resposta.

Aquesta solució no resol completament el problema, però és fàcil d'implementar i pot ser suficient mentre es confia en la cooperació de l'usuari.

- **Segona iteració:** Un altre problema detectat és que les preguntes numèriques poden rebutjar respostes vàlides. Per exemple, per a la pregunta sobre metres quadrats, la resposta hauria de permetre un rang de 30 a 200, però si l'usuari introdueix 30, el sistema demana que respongui de nou. Això és perquè el sistema només accepta valors que es troben a la base de dades i no permet 30 perquè no hi ha cap propietat amb aquesta mida. Aquesta estratègia pot ser útil per a preguntes categòriques, però no és adequada per a les numèriques.

La solució és que el sistema reconegui un rang acceptable de respostes (amb la funció "extract_range") i comprovi si la resposta de l'usuari cau dins d'aquest rang.

Tot i això, el sistema continua rebutjant respostes amb decimals, assumint que l'usuari no donarà respostes tan precises, com sol passar en el mercat real.

- **Tercera iteració:** Les preguntes categòriques només accepten una resposta que coincideixi amb les opcions disponibles. Tanmateix, pot ser que l'usuari vulgui seleccionar més d'una opció, com per exemple viure tant a Barcelona com a Badalona.

La implementació d'aquesta funcionalitat és senzilla. Només cal processar la resposta de l'usuari de manera que es guardin totes les opcions seleccionades en un conjunt. A l'hora de filtrar les

propietats, simplement es comprova si la ubicació de la casa es troba dins del conjunt de preferències de l'usuari. Aquesta nova implementació permet també que l'usuari pugui donar respostes en forma d'oracions, en lloc d'opcions simples.

El principal límit d'aquesta solució és garantir que totes les opcions disponibles dins l'oració siguin correctament interpretades com a preferències. Tot i això, ja s'ha avisat a l'inici que les respostes han de ser clares i concises, de manera que no considerarem aquest possible nou problema.

- **Quarta iteració:** Durant la fase de PROVES, com a desenvolupadors del sistema, si la resposta és rebutjada, podem intentar deduir quin ha estat l'error. Però un usuari no té aquesta capacitat, ja que desconeix completament com funciona el sistema. Per tant, hem decidit que el sistema imprimeixi un missatge indicant quin tipus d'error ha comès l'usuari en la seva resposta, de manera que pugui corregir-la més fàcilment.
- **Cinquena iteració:** En el conjunt de preguntes n'hi ha dues que estan relacionades amb el pressupost de l'usuari: els ingressos mensuals i el pressupost total. Aquesta dualitat pot generar confusió durant el procés de recomanació de propietats.

El problema apareix quan el pressupost de l'usuari supera el 35% dels seus ingressos mensuals, especialment en el cas de preferir un habitatge de lloguer. En iteracions anteriors, quan es produïa aquesta ambigüitat, el sistema donava prioritat a la resposta més restrictiva. Aquesta estratègia, però, pot ser inconvenient si l'usuari està disposat a destinar més d'un 35% dels seus ingressos al lloguer.

La solució implementada consisteix en avisar l'usuari quan el seu pressupost supera aquest límit, recomanant-li que redueixi el pressupost. A continuació, se li pregunta si vol mantenir el pressupost original. Si la seva resposta és afirmativa, el sistema continuarà sense aplicar la restricció del 35% en la cerca d'habitatges. Si la resposta és negativa, es torna a demanar un nou pressupost, iniciant un bucle que es repeteix fins que l'usuari proporcionï un pressupost més ajustat o decideixi ignorar la recomanació del sistema.

3 Conservation flow

El conversation flow és una eina molt potent que ens ajuda a detectar visualment quin és el recorregut de la conversa, a on es dirigeix, quins són els camins feliços i casos límits o fins i tot, l'objectiu del diàleg.

La plataforma amb la qual hem dibuixat el nostre conversation flow és "Voiceflow", però no hem seguit estrictament la seva literatura sinó que hem aprofitat el seu disseny per fer una gràfica més visual que funcional (tal com ho permet). L'enllaç on es troba és (l'enllaç caduca passats les 72 hores, en cas que es passi contacta amb nosaltres per enviar un de nou): https://creator.voiceflow.com/workspace/accept-invite?inviteToken=eyJhbGciOiJIaXciLCJlbnMiOiJBMjU2R0NNIn0..hGFr8KE8e0DNae0j.PVLX3xJBVwLRJ1dFXl4ygDv_yjcumD5wAaZz8NAGVChUqk48vo1yT15Uy3Td03lmmC01gqIz4LXIW0qJBcA0otLzQqgZebG0cbwRamxqBcuD0jpuL6A.HDFA8GwPpalhl0SoPifFLg

En l'esquema, hem representat el flux de la conversa amb tots els camins possibles. En el diagrama:

- Els blocs de color vermell serveixen per transmetre els missatges i rebre les respostes de l'usuari, és a dir, formen el flux principal. Les preguntes poden ser de resposta numèrica o amb opció múltiple.
- Si en la pregunta d'opció múltiple l'usuari respon escollint alguna opció vàlida, es passa al següent bloc. Si la resposta és “quit”, s'acomiada i s'acaba la conversa. En altres casos se salta el “warning” #1 avisant a l'usuari que l'opció que ha respost no és vàlida i torna a demanar resposta.
- Els blocs de color gris serveixen per analitzar les preguntes de resposta numèrica. Tal com s'ha dissenyat, si la resposta no conté números surt el “warning” #2 avisant que a la resposta no hi conté números. Si la resposta surt del rang de valors possibles també hi ha el “warning” #3 al respecte. Si la resposta és “quit”, s'acomiada i s'acaba la conversa. En altres casos es passa al següent bloc.
- Els “warnings” corresponents es troben en la llegenda.
- Cada resposta es guarda en la variable corresponent per representar la variable que en els codis es diu “user_preferences”.
- Per claredat, les línies verdes són els fluxos cap a la fi de la conversa.

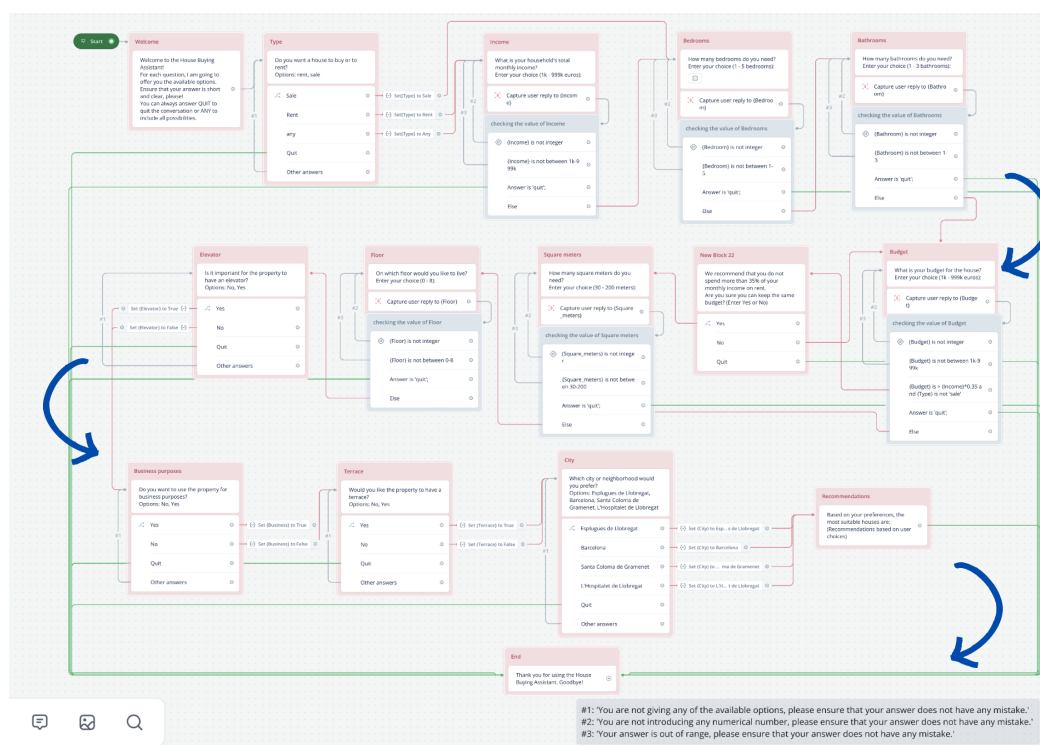


Figura 1: Conversation Flow

Els camins feliços són els escenaris ideals on el xatbot segueix les regles definides sense problemes ni mal interpretacions. El nostre xatbot es romandrà dintre dels camins feliços si l'usuari proporciona

entrades esperades, sense ambigüitat i ben estructurades. Tot i això, el xatbot encara no és prou complex i hi ha casos límit quan: contradiu els requisits anteriors, a les respostes de preguntes d'opció múltiple no s'escriu la paraula clau com hi és predefinida (per exemple “any” i “quit” s'han d'escriure soles i sense context, “Barcelona” no es pot escriure com “BARCELONA”), a les respostes de preguntes numèriques no conté nombres enters o surt del rang possible, els valors amb mil no poden tenir punt (1.5k no vàlid, ha de ser 1500).

4 Conclusió

Aquest treball ha suposat el desenvolupament d'un sistema de diàleg bàsic capaç d'interactuar amb l'usuari per recomanar-li habitatges segons les seves preferències. Al llarg del procés, hem aplicat diferents tècniques i metodologies, com el mètode d'Exemples, el Cas Límit/Camí Feliç, el Prototipat i les Proves, per millorar progressivament el funcionament del sistema i assegurar una experiència d'usuari òptima.

En primer lloc, hem determinat que el diàleg és una eina efectiva per resoldre la tasca de recomanació d'habitatges. La interacció basada en preguntes permet una comunicació directa i personalitzada amb l'usuari, facilitant la recollida de les seves preferències. A través d'un sistema de diàleg dirigit pel propi sistema, hem pogut gestionar les seves respostes de manera estructurada, adaptant les recomanacions en temps real. Pel que fa a la personalitat del sistema, hem optat per un enfocament funcional i guiat, oferint instruccions clares i senzilles que fan més accessible l'ús del sistema.

La implementació es basa en el llenguatge Python, on la classe ‘Chatbot’ és responsable de la gestió del diàleg, emmagatzematge de dades, accés a la base de dades, la presa de decisions en funció de les preferències de l'usuari i la generació d'un llenguatge adequat per interactuar amb l'usuari. Al mateix temps, altres funcions auxiliars tracten la comprensió del llenguatge natural, assegurant que les respostes numèriques i categòriques siguin tractades correctament.

Al llarg de diverses iteracions, hem abordat limitacions com la comprensió de respostes numèriques i categòriques, ampliant el rang de respostes possibles i oferint retroalimentació útil a l'usuari per corregir errors. Així, el sistema ha evolucionat per filtrar millor les propietats i ajustar-se millor a les expectatives de l'usuari. L'enfocament iteratiu ha estat clau per assegurar millores constants en la gestió del diàleg i la precisió del sistema.

En conclusió, aquest projecte demostra com l'ús de metodologies iteratives i tècniques de gestió del diàleg pot resultar en un sistema que, tot i ser senzill, proporciona una experiència d'usuari efectiva. Encara hi ha espai per millores, especialment en la comprensió i generació del llenguatge natural, amb l'objectiu de crear un sistema més sofisticat que pugui gestionar converses més complexes de manera fluida i natural.