Obiettivo

La realizzazione di un chatbot generalmente, impiega due figure professionali: la prima si occupa di disegnare il flusso di dialogo tra chatbot stesso e utente finale, implementando le frasi legate al contesto conversazionale, le risposte, le parole chiave di rilievo.

La seconda invece, si occupa di realizzare il software vero e proprio finalizzato all’implementazione del chatbot vera e propria.

Questo processo, richiede parecchio tempo e un continuo confronto tra le due figure prese in esame: generalmente, chi si occupa di disegnare i flussi di dialogo non ha esperienza o abilità legate alla programmazione, ha comunque conoscenze informatiche limitate che non gli permettono di realizzare software autonomamente.

Al contempo, chi ha notevoli competenze informatiche non è generalmente una figura professionale che possa realizzare flussi di dialogo in autonomia.

Il nostro obiettivo, è di fatto fornire un sistema di semplice utilizzo per i disegnatori di flussi di dialogo, affinché possano in totale autonomia realizzare un chatbot da zero senza saper programmare.

Stato dell’arte

Il mercato offre numerose soluzioni di questo genere, tuttavia hanno diverse lacune.

Nella totalità dei casi, non prevedono la possibilità di istruire il bot. In molti casi gli utenti finali commettono errori di sintassi, questo implica che il chatbot non sia in grado di carpire ciò che l’utente vuole comunicare, trovandoci a un bivio.

La prima strada, consiste nel realizzare un numero non indifferente di condizioni affinché il bot riconosca gli errori di battitura, di sintassi o frasi sconnesse dal contesto conversazionale: è un lavoro non indifferente che richiede un notevole quantitativo di tempo.

La seconda soluzione invece, è quella di far rientrare tutti questi problemi in un unico stato dove il bot invita l’utente a ripetere ciò che desidera comunicare (rispondendo ad esempio con “Non ho capito, potrebbe ripetere per favore?”).

Un’altra limitazione di cui molti framework soffrono, è legata alla piattaforma d’uso. Spesso, si tratta di chatbot da usare per le pagine Facebook, widget per siti web o altri social network che permettono conversazioni tra utenti.

Per tanto, l’utente si vede costretto a usare una specifica piattaforma senza troppa libertà di scelta.

In altri casi, si è limitati a una conversazione “botta e risposta”: non esiste un vero flusso di dialogo. L’utente ha la possibilità di cliccare su bottoni, oppure scrivere frasi ben specifiche dettate dal chatbot stesso. Non si va a creare un flusso di dialogo vero e proprio, bensì un botta e risposta tra utente e chatbot.

In diversi casi invece, è necessario interrogare API esterne per offrire all’utente i dati di cui ha bisogno.

Nella totalità dei casi, è un servizio non previsto dai framework presenti a mercato, ad eccezione di pochi casi in cui non si ha troppo margine di manovra, impedendo al designer del chatbot di impostare i parametri necessari a invocare le API e gestire le eventuali risposte con i parametri ad esse associate (ovvero, i dati da offrire all’utente finale).

Dialogflow invece, è un servizio di Google che permette la realizzazione di chatbot offrendo non pochi strumenti per un’implementazione a 360 gradi:

* Machine learning: permette l’auto miglioramento del chatbot in funzione di ciò che gli utenti scrivono. Questo significa che possiamo comprendere frasi sgrammaticate, scritte male o in modo parziale.
* Svincolo dalle piattaforme: i chatbot realizzati con Dialogflow permettono di essere usati sia come soluzione standalone, sia come integrazione per servizi di messaggistica (Telegram, Slack, Facebook Messenger, ecc.), sia come servizio aggiuntivo per siti web o applicazioni di terze parti quali Skype, sia come assistente vocale.   
  Quest’ultimo punto è di grande rilievo poiché permette di essere utilizzato con la voce. Dialogflow dà la possibilità di essere abbinato ad assistenti vocali quali Google Home, Alexa e non solo: siamo in grado di integrarlo a sistemi “fai da te” quali Raspberry Pi o altre piattaforme embedded. Tutto ciò permette di fornire supporto a persone con problemi visivi.
* Interrogazione di API esterne: Dialogflow, previa attivazione di un abbonamento a consumo, permette di integrare in totale libertà l’uso di API esterne. Questo permette di integrare al nostro chatbot informazioni legate a servizi di terze parti, quali ad esempio servizi metereologici, di trasporto e dati legati all’ente stesso che intende usarlo.
* Flusso di dialogo libero: siamo in grado di realizzare un flusso di dialogo in totale libertà. Partendo da frasi ipotetiche che l’utente finale può scrivere, possiamo carpirne informazioni di rilievo (parole, frasi, numeri, codici, date, ecc.) in totale libertà. In funzione del flusso di dialogo che stiamo realizzando, abbiamo piena autonomia e totale controllo sulle frasi scritte dall’utente, termini chiave e risposte che il chatbot deve fornire di conseguenza.

Questo framework tuttavia, ha due limitazioni.

Pur offrendo un’interfaccia grafica abbastanza lineare, un utente con scarse conoscenze di programmazione può trovare difficoltà nell’implementazione del chatbot.

Nel caso in cui desideri implementare chiamate ad API esterne, ha la necessità di saper programmare in Javascript: un chatbot realizzato in Dialogflow ha ad esso associato un file JS che prevede eventuali invocazioni ad API esterne in cui va gestita la risposta e gli eventuali dati forniti.

Soluzione

La soluzione per far fronte a ciò, è dare la possibilità di disegnare il chatbot stesso tramite un sistema puramente visuale, impiegando DrawIo. L’utente, tramite un linguaggio ad eventi ad hoc, ha la possibilità di realizzare un grafo seguendo delle linee guida rendendogli totalmente trasparente il processo di costruzione del chatbot, eliminando i passi in cui la programmazione era precedentemente necessaria.

Questo processo è possibile come segue: una volta realizzato il grafo che rappresenta il flusso di dialogo del chatbot con le eventuali chiamate ad API esterne, ne si realizza un file XML che verrà interpretato dal software di mia realizzazione.

Una volta terminato il processo di interpretazione del file, estraendo tutte le informazioni di rilievo per la realizzazione del chatbot, vengono impiegate le API di Google per poter dare vita al chatbot.

Questo processo di interpretazione e realizzazione del chatbot è totalmente trasparente all’utilizzatore finale: potrà visionare in automatico il chatbot stesso accedendo al suo account di Dialogflow per poi poterlo impiegare in base alle sue necessità.

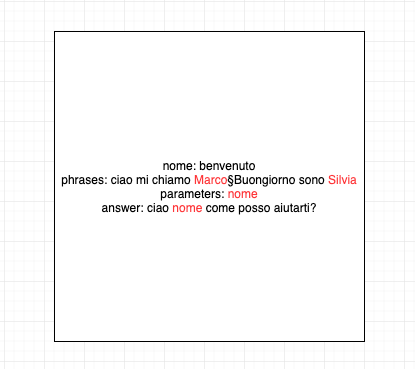
Linguaggio visuale

Il sistema di realizzazione dei bot ha come obiettivo quello di guidare qualsiasi utente, specialmente coloro che non hanno esperienze di programmazione.

Pur avendo questo obiettivo finale, è necessario specificare delle direttive d’uso, motivo per cui ho implementato un linguaggio visuale strutturato per ovviare alle lacune degli utilizzatori finali.

**Intent**:

L’intent viene costruito con la figura del quadrato con al suo interno il seguente testo:



Ogni intent è costituito da un nome, da delle frasi, da dei parametri e da delle risposte.

La sintassi impiegata è la seguente:

* “nome: NomeDellIntent”: specifica il nome dell’intent stesso
* “phrases: frase1§frase2^fraseN”: sono le frasi ipotetiche che può scrivere l’utilizzatore finale del chatbot. Al loro interno, come si può notare dall’esempio, ci sono alcuni termini con una determinata colorazione, sulla quale torneremo in seguito. Per poter utilizzare più di una frase, è sufficiente usare come carattere separatore “§”.
* “parameters: parametro1§parametro2§parametroN”: sono i parametri. L’intent, una volta ascoltato l’utilizzatore, carpisce dalle frasi scritte le informazioni di rilievo scritte dall’utilizzatore stesso affinchè si possano usare nel flusso della conversazione. Così come per le frasi, anche in questo caso si utilizza il carattere “§” come separatore nel caso in cui si desideri inserire più di un parametro all’interno dell’intent.
* “answer: risposta1§risposta2§rispostaN”: così come per parametri e frasi, anche in questo caso si utilizza il carattere separatore “§” per distinguerle. Queste, sono le risposte che darà il chatbot nel momento in cui la conversazione si trovi in questo specifico intent.

**I colori**: riferendoci sempre all’esempio precedente, si può notare come parametri, frasi di ingresso e risposte, abbiano delle parole colorate.

Tali colorazioni servono ad associare il parametro coinvolto (che nel caso dell’esempio è il parametro “nome”) con le parole contenute nelle frasi e nelle risposte.

Questo passo è fondamentale affinché il chatbot possa mantenere memorizzato ciò che ha scritto l’utente di rilievo (nell’esempio, il suo stesso nome), per poterlo usare nel flusso conversazionale (nell’esempio, il chatbot potrà rispondere all’utente per nome, chiedendo come può aiutarlo).

Andando nel concreto, qual ora l’utente scriva “Ciao mi chiamo Andrea”, il chatbot capirà che la parola “Andrea” contenuta nella frase, andrà a valorizzare il parametro “nome” per poi in seguito rispondere all’utente con “Ciao Andrea come posso aiutarti?”.