

# Un Agente Dialogante in Ambito Finanziario per il progetto UWMP

Fedelucio Narducci, Andrea Iovine, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro<sup>1</sup> and Georgios Lekkas, Nicola Valenza<sup>2</sup>

<sup>1</sup> Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, nome.cognome@uniba.it

<sup>2</sup> Objectway s.p.a., nome.cognome@objectway.com

## Abstract

L'iniziativa UNIFIED WEALTH MANAGEMENT PLATFORM (UWMP) ha l'obiettivo di implementare un nuovo modello di servizio nel mondo delle istituzioni finanziarie che supera il modello del consulente fisico presente in agenzia. La piattaforma vuole offrire ai clienti e utilizzatori dei prodotti finanziari un'ampia disponibilità dei servizi di investimento, diversificati e 'assemblabili' sulla base delle proprie esigenze, con capacità di utilizzo autonoma e senza intermediazioni. Tra le varie attività di progetto è prevista la progettazione e implementazione di un agente dialogante per lo svolgimento di funzioni informative e dispositive in ambito finanziario. In questo articolo vengono descritte le possibili soluzioni che sono state prese in considerazione per lo sviluppo del suddetto sistema, descrivendo per ognuna le caratteristiche salienti, vantaggi e svantaggi.

## 1 Introduzione

Gli assistenti conversazionali goal-oriented possono essere classificati in tre categorie:

- **Informativi:** la tipologia più semplice di chatbot, i quali hanno lo scopo di fornire informazioni generali come ad esempio FAQ, notizie o notifiche.
- **Dispositivi:** permettono agli utenti di eseguire transazioni (ad esempio, prenotare un hotel, inviare un bonifico).
- **Consulenti:** sono dotati di capacità di apprendere, e possono offrire ulteriori prodotti e servizi sulla base delle transazioni e conversazioni precedenti.

Date le caratteristiche del progetto UWMP, il cui scopo è quello di implementare un agente dialogante in grado di svolgere funzioni informative e dispositive in ambito finanziario, si è deciso di implementare un sistema conversazionale goal-oriented di tipo modulare. Un agente conversazionale goal-oriented [1] è solitamente composto da una serie di componenti, ciascuna con una particolare responsabilità [2]. Nella prossima Sezione verrà presentata l'architettura per l'implementazione dell'agente dialogante nell'ambito del progetto UWMP.

## 2 Descrizione dell'agente dialogante

Una generica architettura di un agente dialogante goal-oriented è definita in Figura 1.

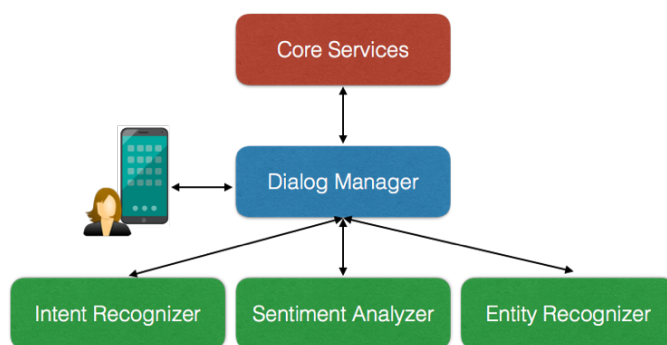


Figura 1: L'architettura dell'Agente Dialogante

L'utente interagisce con il sistema tramite un computer o un dispositivo mobile, di solito tramite un'applicazione Web o una piattaforma di messaggistica (come Telegram o Facebook Messenger). Le componenti principali di un agente conversazionale goal oriented sono:

- **Intent Recognizer:** riconosce l'azione o la richiesta effettuata dall'utente;
- **Sentiment Analyzer:** riconosce le emozioni o il sentimento generale espressi dall'utente. Ad esempio: "Sono molto contento", o "Non mi piace". È facoltativo poiché non è richiesto da nessuno dei requisiti del progetto.
- **Entity Recognizer:** riconosce le menzioni a entità come oggetti, persone, organizzazioni, numeri o nomi.
- **Dialog Manager:** mantiene lo stato corrente della finestra di dialogo con l'utente e orchestra le altre componenti per far avanzare il suddetto stato. Ad esempio, dopo aver ricevuto dall'utente il messaggio "Invia un pagamento di 200 EUR a Bob Jones", il Dialog Manager inoltrerà questo messaggio all'Intent Recognizer che riconoscerà l'intenzione di avviare un bonifico bancario e all'Entity Recognizer che invece estrarrà il beneficiario (Bob Jones) e l'importo (200 EUR). Infine, comunicherà con la componente Core Services per avviare la procedura di pagamento.

### 3 Decisioni progettuali

La fase preliminare di analisi ha avuto l'obiettivo di scegliere gli strumenti più adatti per l'implementazione dell'agente conversazionale. Le opzioni sono state valutate in base ai requisiti richiesti dal progetto. L'Entity Recognizer deve essere in grado di riconoscere nomi di entità del dominio (e di associarli ad un identificativo univoco), numeri, date, nomi di persone e una serie di parole chiave predefinite. Oltre a questo, la componente deve essere in grado di riconoscere le espressioni temporali, per esempio: "nell'ultimo mese", "dall'1 Gennaio al 31 Agosto". Il Dialog Manager deve essere in grado di offrire una funzionalità di slot-filling, per chiedere all'utente tutte le informazioni necessarie in maniera ordinata e flessibile. Deve inoltre effettuare controlli sui parametri ricevuti e riempire automaticamente alcuni parametri in base a delle condizioni interne. Infine, deve essere in grado di gestire disambiguazioni quando necessario.

Per ridurre il costo di sviluppo di un agente dialogante sono stati valutati diversi framework. In particolare, sono state prese in considerazione due piattaforme: DialogFlow<sup>1</sup> e Watson Assistant<sup>2</sup>. Entrambe mettono a disposizione un ambiente grafico per la definizione del modello di dialogo, sul quale è possibile definire gli intent che possono essere riconosciuti in ogni momento della conversazione. Inoltre, forniscono delle basilari funzionalità di Intent Recognition, Entity Recognition e Dialog Management, con capacità di slot-filling. Tuttavia, nessuna delle due è in grado di coprire da sola tutti i requisiti necessari, sia per la componente di Entity Recognition, sia per quella di Dialog Management.

Per ragioni di licensing, si è deciso di prelevare da DialogFlow la componente di Intent Recognition, che offre una sufficiente funzionalità per lo scopo del progetto.

Per quanto riguarda il Dialog Manager, e in particolare la funzione di slot-filling, si è scelto di non adattare quelle fornite dalle due piattaforme, e di sviluppare una soluzione *ex novo*, che supporti nativamente i requisiti del progetto e permetta di ridurre la dipendenza da piattaforme commerciali.

Le funzionalità di Entity Recognition necessarie possono essere realizzate tramite l'uso della libreria Stanford CoreNLP<sup>3</sup>, che permette il riconoscimento di entità, numeri, date ed espressioni temporali, e di una libreria di Fuzzy Search, che permette il collegamento delle entità con un identificativo univoco (un esempio di libreria di questo tipo è Apache Lucene<sup>4</sup>).

### 4 Funzionalità dell'agente dialogante UWMP

L'agente dialogante sviluppato nel corso di questo progetto ha l'obiettivo di supportare i clienti di banche o altri istituti finanziari nello svolgimento delle operazioni generalmente effettuate allo sportello o mediante siti web. L'uso del linguaggio naturale cerca di ridurre lo sforzo e la complessità dell'interazione. Il bot agisce in maniera reattiva, ossia risponde a

richieste dell'utente, scritte in lingua inglese. Il sistema sviluppato è in grado di rispondere a richieste sia di carattere informativo, sia di natura dispositiva. Ad esempio, può fornire informazioni come il saldo di un conto corrente, il valore di un portfolio di azioni, le performance dei propri investimenti e rispondere ad eventuali quesiti generici. Per conoscere il saldo del conto, l'utente può porre la domanda: *"What is my cash account balance?"*. Il sistema fornirà le informazioni richieste, rispondendo: *"Your cash account balance is 4890,00 EUR"*. L'utente può tenere sotto controllo l'andamento dei propri investimenti. È possibile avere un resoconto generale o per uno specifico intervallo di tempo. Alcuni esempi di richieste di questo tipo sono: *"What was my portfolio performance in the last three weeks?"*, *"What was my performance from January 1st to August 31st?"*. In questi esempi, le componenti che intervengono per completare il task sono l'Intent Recognizer, che dovrà riconoscere l'azione richiesta, l'Entity Recognizer che dovrà individuare l'espressione temporale e, naturalmente, il Dialog Manager che funge da orchestratore. Come detto in precedenza, l'agente è anche abilitato ad eseguire transazioni finanziarie, in particolare, bonifici e investimenti. L'utente dovrà fornire tutte le informazioni necessarie per l'esecuzione del task attraverso messaggi in linguaggio naturale (secondo il modello di slot-filling su descritto). Per evitare la richiesta di informazioni già in possesso dell'agente, viene gestito un database che memorizza, ad esempio, i beneficiari dei precedenti bonifici, individuabili tramite nome o un altro alias. L'utente potrebbe dunque scrivere il messaggio: *"Send a payment of 200 EUR to Bob Jones"*. Nel caso il beneficiario sia già noto, l'agente sarà pronto ad effettuare il trasferimento di denaro, rispondendo: *"I prepared a payment of 200 EUR to Bob Jones, alias Uncle Bob, at Barclays account..."*. Alternativamente, l'utente potrebbe semplicemente chiedere: *"I want to make a payment"*. In questo caso l'agente chiederà: *"What is the name of the payee?"*. Ricevuta la risposta, il sistema chiederà altre informazioni come la somma da pagare o il codice IBAN, finché non avrà le informazioni sufficienti per completare l'operazione. In maniera analoga vengono gestite altre operazioni dispositive come gli investimenti. Anche in questo caso, l'utente può semplicemente indicare: *"I want to make an investment"*. Il sistema dovrà dunque chiedere quale tipo di investimento eseguire, ponendo la domanda: *"Do you want to invest in stocks or funds?"*. Il riconoscimento di azioni o fondi nei messaggi tollera errori di battitura o menzioni incomplete grazie alla flessibilità dell'Entity Recognizer. Nel caso più prodotti siano compatibili con quello menzionato dal cliente, l'agente gestisce la disambiguazione ponendo ulteriori domande.

### Riferimenti bibliografici

- [1] K. Branting, J. Lester, and B. Mott. Dialogue management for conversational case-based reasoning. In *European Conference on Case-Based Reasoning*, pages 77–90. Springer, 2004.
- [2] J. Williams, A. Raux, and M. Henderson. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33, 2016.

<sup>1</sup><https://dialogflow.com/>

<sup>2</sup><https://www.ibm.com/cloud/watson-assistant/>

<sup>3</sup><https://stanfordnlp.github.io/CoreNLP/>

<sup>4</sup><http://lucene.apache.org/>