A Virtual Customer Assistant for the Wealth Management domain in the UWMP project

Andrea Iovine
Università degli Studi di Bari Aldo Moro, Bari, Italy
andrea.iovine@uniba.it

Marco de Gemmis
Università degli Studi di Bari Aldo Moro, Bari, Italy
marco.degemmis@uniba.it

Fedelucio Narducci
Università degli Studi di Bari Aldo Moro, Bari, Italy
fedelucio.narducci@uniba.it

Giovanni Semeraro
Università degli Studi di Bari Aldo Moro, Bari, Italy
giovanni.semeraro@uniba.it

Doriana Filisetti
Objectway SpA, Milan, Italy
doriana.filisetti@objectway.com

Davide Ingoglia
Objectway SpA, Milan, Italy
davide.ingoglia@objectway.com

Georgios Lekkas
Objectway SpA, Milan, Italy
georgios.lekkas@objectway.com

ABSTRACT

The Universal Wealth Management Platform (UWMP) project has the objective of creating a new service model in the financial domain. An integral part of this service model is the creation of a new Virtual Customer Assistant, that is able to assist customers via natural language dialogues. This paper is a report of the activities performed to develop this assistant. It illustrates a general architecture of the system, and describes the most important decisions made for its implementation. It also describes the main financial operations that it is able to assist customers with. Finally, it delineates some avenues for future work.

CCS CONCEPTS

• **Human-centered computing** → **Natural language interfaces**; • **Computing methodologies** → *Information extraction*; *Discourse, dialogue and pragmatics.*

KEYWORDS

Conversational Agents, Digital Assistants, Finance, Wealth Management

ACM Reference Format:
Andrea Iovine, Marco de Gemmis, Fedelucio Narducci, Giovanni Semeraro, Doriana Filisetti, Davide Ingoglia, and Georgios Lekkas. 2020. A Virtual Customer Assistant for the Wealth Management domain in the UWMP project. In *IUI '20 Workshops, March 17, 2020, Cagliari, Italy*. ACM, New York, NY, USA, 4 pages.

1 INTRODUCTION AND BACKGROUND

The purpose of Wealth Management firms is to advise clients on investment strategies, execute orders on their behalf and help them custody their financial assets. This type of work has always been closely associated to personal relationships and confidentiality. Today however, we expect that the coming of age of a new, digital-savvy generation of wealthy individuals will change the relations between clients and wealth management firms.

Digital Assistants such as Amazon Alexa and Apple Siri have popularized the notion of software applications helping users with everyday tasks. In the realm of business, such applications are also known as Virtual Customer Assistants and they support text or voice interactions to deliver information or to act on behalf of the customer. The goal of a Virtual Customer Assistant is twofold: first, to take over routine interactions so that human service agents can engage in more value-adding activities; second, to improve customer experience by reducing friction, i.e. eliminating any factor that can deter or slow down interactions between clients and firm. There are more than 200 software companies offering Digital Assistants for customer interaction. In the financial domain some prominent cases are IBM Watson, Microsoft Virtual Assistant, boost.ai, Creative Virtual. Most of these assistants interact via the bank's mobile app and web site, while a few in the USA use Facebook Messenger.

For example, Royal Bank of Canada has a chatbot called Arbie that helps users open a new account. Bank of America's assistant called Erica [1] interacts with customers using voice and text messages. Wells Fargo's chatbot [10] provides account information via Facebook Messenger. Capital One's chatbot named Eno [3], interacts via SMS to execute money transfers and supply account information.

Objectway is a global Fintech 100 software provider of the wealth, investment & asset management industry. The company launched its Universal Wealth Management Platform (UWMP) initiative to implement a new, fully-digital service model destined to financial institutions. The new digital model complements the existing model based on financial advisors and physical interactions, by offering clients self-serve functionality on a wide range of investment services.

One of the activities of the UWMP project in support of the new digital service model was to implement a service that enables clients to interact with wealth management firms using Natural Language dialogue. The goal was to empower clients with informational and transactional capabilities that were always-on and accessible from everywhere. The new service would have to satisfy the requirement of confidentiality, so priority was set on text interactions. Work on voice was postponed due to potential issues with revealing sensitive client information. Another requirement was to ensure high quality of interactions, since failures would have an adverse effect on the perception of quality and trust that clients place on wealth management firms. This advised us to focus on the most recurring, time-consuming routine tasks that clients faced when interacting with the firm, such as requesting information on account balance and performance.

To try out more complex dialogues, we also decided to explore tasks of a transactional nature, although we expect the adoption of such features in real life to be slow. Investment transactions increase the difficulty of interaction due to the number and variety of financial products, order types and the associated ambiguity that derives from such complexity. The rest of this paper is organized as follows: the system architecture is describe in Section 2, while Section 3 shows how the system works. Conclusions are outlined in Section 4.

2 SYSTEM ARCHITECTURE

Chatbots can be classified as *open-domain* or *goal-oriented*. Open-domain agents handle generic conversations, while goal-oriented chatbots interact with users via natural language conversations to assist them in their tasks. Goal-oriented CAs can then be classified as informational, transactional or advisory. Advisory chatbots learn from users, and can recommend products based on the interaction, such as in [6, 7]. Our digital assistant falls into the category of *goal-oriented* chatbots [2]. Figure 1 shows a typical architecture for a goal-oriented conversational agent.

This architecture is called *modular*, as it is made up of several components, each with a specific task.

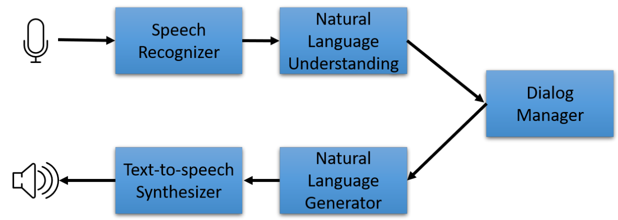

Figure 1: Architecture of a goal-oriented Conversational Agent (adapted from [11, 12])

The *Speech Recognizer* component has the responsibility of transcribing the user's voice input into a text message. This is an optional component used by conversational agents that support a voice-based interaction.

The *Natural Language Understanding* (NLU) component takes in input the user's text message, and outputs a semantic representation of that message. The objective is to understand the meaning of the message itself. This means performing several Natural Language Processing tasks. In the NLU component, *Intent Recognition* is done to understand the action or the request that the user is making. For example, the message *"Pay 200 USD to Alice"* means that the user is trying to start a payment. *Entity Recognition* is also a common NLU task, which is used to recognize mentions to entities such as people, organizations, or numbers. From the previous example message, it should be able to extract *"200 USD"* as a monetary amount, and *"Alice"* as a person name. *Sentiment Analysis* is performed to understand the emotional state expressed through the message. This could be used to make the agent react differently to different emotional states. For example, a customer support agent can decide to hand off the request to a human operator, in case it detects that the user is frustrated.

The *Dialog Manager* (DM) component maintains an estimate of the current state of the conversation. Based on this state and the current user message, it decides the next appropriate action to perform in order to complete the user's request. For example, the DM might respond by asking a question (if some information is needed), or by performing a transaction (if all information has been provided). Typically, Goal-Oriented agents employ a *frame-based* Dialog Manager [8], in which goal-specific data is enclosed into frames, and can be provided via *slot filling*. For example, the payment frame requires several slots, such as the name of the payee, the amount, and the type of payment. In order to complete a payment, the DM will need to ask these slots one by one. Users are also free to provide the values to the slots in any preferred order, and the DM reacts accordingly. In our architecture, the Dialog Manager also interacts with external services that manage the domain-specific data and perform the transactions. A *Natural Language Generator* (NLG) component is used to transform the output of the DM into a textual form, which can then be presented to the user. In our architecture, the NLG component uses template responses, that are filled dynamically with domain-specific data, as done in Shah et al. [9]. Finally, voice-based systems adopt a *Text-to-speech synthesizer* component to transform the text response of the NLG into speech.

Our agent implements the NLU, DM and NLG, with Intent and Entity recognition functionalities implemented in the NLU component. An analysis phase was conducted in order to determine the most appropriate tools for developing the aforementioned components. Several options were vetted, based on their ability to cover the requirements of the project. In particular, we opted for solutions that require little training, since traditional supervised methods require large amount of domain-specific conversational corpora, such as [5]. Big tech companies such as Google or IBM already provide their own platforms for the development of conversational agents, such as Dialogflow[1] or Watson Assistant[2]. These platforms provide a simple implementation for most of the components of the architecture described in Figure 1, such as Intent Recognition, Entity Recognition, and Dialog Management. However, due to several limitations, not all of their components are suitable for the requirements of our project. Therefore, we decided to adopt the Intent Recognition component from Dialogflow, while the other parts of the architecture are custom-made.

The Entity Recognition (ER) module used in the system is implemented using Stanford CoreNLP[3] and Apache Lucene[4]. It is able to do both Named Entity Recognition and Entity Linking. The ER function can recognize entities such as numbers and person names, and also mentions to entities that are contained in a dictionary (e.g. financial products). It uses a combination of classifiers in order to recognize entity mentions in the user message. A *regex* (regular expression) classifier is used to recognize entities and keywords using exact matching. Additionally, it uses a text classifier based on *Conditional Random Fields* (CRF) [4]. The strength of the CRF classifier is that it exploits the sentence structure to detect entity mentions. Therefore, it is able to recognize entities that cannot be confined in a dictionary (such as person names), as well as misspelled and incomplete entities. The output of the Entity Recognition function is then passed to the Entity Linking function. Entity linking is used to map certain entity mentions to objects in a knowledge base. Fuzzy matching is used for the entity linking step, which allows the ER component to map misspelled and incomplete entities. The ER component can also recognize dates such as "November 13th, 2019", or relative time expressions such as "three weeks ago", using the CoreNLP SUTime component.

The Dialog Manager component was also custom-made for this project. As said before, it uses frames to store the goal-specific information, and slot filling to acquire said information during the dialog. Given a user message, the Dialog Manager can either: perform a transaction, request a slot, or request a disambiguation. Disambiguation is required when multiple entities are found as possible candidate values for a slot. For example, given the user utterance *"I want to buy 200 BMW shares"*, the ER component may recognize two different products: "BMW AG" and "BMW ORD". Therefore, the user will be asked to clarify which one should be selected. During the conversation, the Dialog Manager also performs checks on the slot values, and can auto-fill slot values based on some conditions (e.g. when the user makes a payment to a known payee).

3 THE SYSTEM AT WORK

The main objective of the Virtual Customer Assistant created for the UWMP project is to assist customers of banks or other financial institutions. The bot is reactive, that is, it responds to requests made by users. It can respond both to information requests, and can perform financial transactions. This means that it can be categorized both as a informational and transactional agent.

From the informational side, the agent can be used to obtain general information about the user's financial situation, such as the balance of the user's cash account. Figure 2 shows an example of this use case. Given the message *"What is my cash account balance?"*, the NLU component will recognize the *account balance* intent, and the agent will provide the requested information, responding with: *"Your cash account balance is 4890.00 EUR"*. Users can also keep track of the performance of their investments, such as the value, the performance, and the *profit and loss*. The agent can also provide a report for a specific time period provided by the user. For example, a user can type *"What was my portfolio performance in the last three weeks?"*, or *"What is my portfolio performance from May 1st to August 31st?"*. This is possible thanks to the time expression recognition component of the Entity Recognizer, described in Section 2.

Users can also perform financial transactions directly from the chatbot, as said earlier. The most notable transactions are payments and investments. Indeed, users can send payments by writing for example *"Send a payment of 200 USD to Alice Smith"*. In this case, the NLU component will recognize the *payment* intent, and some of the required entities, such as the payee "Alice Smith", and the amount of "200 USD". Alternatively, the user may not provide any information at

[1] https://dialogflow.cloud.google.com/
[2] https://www.ibm.com/cloud/watson-assistant
[3] https://stanfordnlp.github.io/CoreNLP/
[4] https://lucene.apache.org/

the start, e.g. by saying *"I want to make a payment"*. The Dialog Manager reacts to this accordingly, by collecting all the mentioned entities, and prompting the user for the missing information via slot filling. This can be seen in the example in Figure 2. In this case, the user provided the payee, but not the amount. Thus, the Dialog Manager responds by asking: *"Please tell me the amount to transfer"*. If the user wants to make a payment to an already known payee, some slots will be automatically filled, making repeated payments easier. If multiple payees are found, the Dialog Manager will prompt the user to specify the correct one. When all the required information is collected, the Dialog Manager will ask for a final confirmation, e.g. by saying *"I prepared a payment of 200 USD to Alice Smith, alias Aunt Alice, at Barclays account..."*. An authorization code is required to complete the transaction, to increase security. When the correct code is entered, the agent completes the transaction, and returns a positive feedback to the user, as seen in Figure 2.

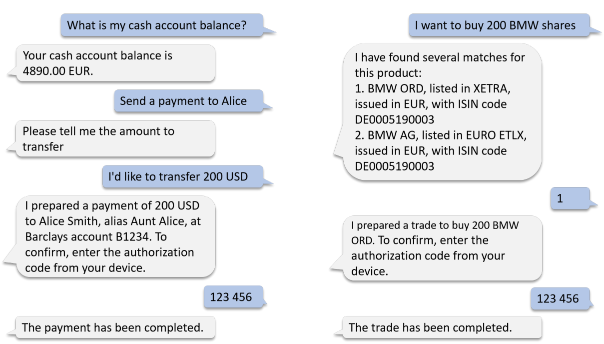

Figure 2: Examples of conversation with the chatbot

The agent also lets the user invest in funds or trade stocks. Investments can be made by specifying a product, the number of stocks (or the amount of money to invest), and the direction (buy or sell). For example, given the message *"I want to buy 200 BMW shares*, the *investment* intent will be recognized by the NLU component, as well as the amount "200", and the "buy" keyword, indicating the direction of the trade. The "BMW" entity will be extracted by the Entity Recognizer, and the Entity Linking component will try to match it to two different products: "BMW AG" and "BMW ORD". In this case, the Dialog Manager will request the user to specify the correct one, e.g. by saying *"I found several matches for this product."*. This can be seen in the example in Figure 2. Just like for payments, missing information is acquired via slot filling.

4 CONCLUSION

In this paper, we presented a Virtual Customer Assistant developed for the UWMP project. The aim of the project is to assist users in completing financial operations of varying complexity using natural language dialogues. The agent supports a wide range of informational and transactional operations, and several components were developed to support them. As future work, we plan to investigate the introduction of a voice-based interaction model. Another avenue for future work is the introduction of advisory functionalities, such as financial product recommendations, that enable the agent to take a more proactive role in the interaction. Finally, the Virtual Customer Assistant will be subject of in-vitro and in-vivo experiments, that will assess its ability to aid users in their financial tasks.

5 ACKNOWLEDGEMENTS

This work has been funded by the project UNIFIED WEALTH-MANAGEMENT PLATFORM - OBJECTWAY SpA - Via Giovanni Da Procida nr. 24, 20149 MILANO - c.f., P. IVA 07114250967.

REFERENCES

[1] Bank of America Erica [n.d.]. Meet Erica, Your Financial Digital Assistant From Bank of America. https://promo.bankofamerica.com/Erica/
[2] Antoine Bordes, Y.-Lan Boureau, and Jason Weston. 2016. Learning End-to-End Goal-Oriented Dialog. *arXiv:1605.07683 [cs]* (May 2016).
[3] CapitalOne Eno [n.d.]. Eno, your Capital One assistant. https://www.capitalone.com/applications/eno/
[4] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. 363–370.
[5] Andrea Iovine, Fedelucio Narducci, and Marco de Gemmis. 2019. A Dataset of Real Dialogues for Conversational Recommender Systems. In *CLiC-it 2019*. 6.
[6] Fedelucio Narducci, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2019. An investigation on the user interaction modes of conversational recommender systems for the music domain. *User Modeling and User-Adapted Interaction* (Nov. 2019).
[7] Fedelucio Narducci, Pierpaolo Basile, and Andrea Iovine. 2018. A domain-independent Framework for building Conversational Recommender Systems. In *KARS@Recsys*. 29–34.
[8] A Rudnicky. 1999. AN AGENDA-BASED DIALOG MANAGEMENT ARCHITECTURE FOR SPOKEN LANGUAGE SYSTEMS. *IEEE Automatic Speech Recognition and Understanding Workshop* 13 (1999), 4.
[9] Kiner B Shah. 2017. Approaches towards Building a Banking Assistant. *International Journal of Computer Applications* 166 (2017), 6.
[10] Wells Fargo Chatbot 2017. Wells Fargo first U.S. bank with Facebook Messenger chatbot. https://stories.wf.com/helpful-banking-assistanton-facebook/
[11] Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The Dialog State Tracking Challenge Series: A Review. *Dialogue & Discourse* 7, 3 (April 2016), 4–33.
[12] Tiancheng Zhao and Maxine Eskenazi. 2016. Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning. (Jun 2016), 1–10.