

Implementing a Pragmatically Adequate Chatbot in DialogFlow CX

Anna Dall’Acqua^{1,2}, Fabio Tamburini¹

1. FICLIT, University of Bologna, Italy

2. Injenia S.r.l., Bologna, Italy

anna.dallacqua2@unibo.it, fabio.tamburini@unibo.it

Abstract

This paper presents work in progress concerning the implementation of a list of linguistic patterns developed in an original way to be pragmatically adequate. These patterns for Italian are strongly rooted in Conversation Analysis and are adaptable and portable into different domains. The platform used for the implementation is Dialogflow CX.

1 Introduction

Although the first dialogue systems began to appear around the second half of the last century (Weizenbaum, 1966; Colby et al., 1971) is it especially in recent years that we have witnessed a proliferation of these technologies in a wide variety of fields (Tsvetkova et al., 2017; Chaves et al., 2019; Dale, 2016). The numerous attempts that have been made to classify them (Radziwill and Benton, 2017; Følstad et al., 2019; Hussain et al., 2019; Mathur and Sing, 2018) and the absence of an unequivocal taxonomy (Braun and Matthes, 2019) contribute to the lack of a methodological approach for designing conversational agents.

The recent technological developments have led to the standardisation of the technical frameworks: the main Natural Language Understanding (NLU) platforms, both developed by technology giants such as Google Dialogflow, IBM Watson, and Microsoft Luis and those from the open source community such as RASA, contributed to the affirmation of the dominant paradigm based on *intents*, *entities* and *responses* for building conversational agents (Adamopolou and Moussiades, 2020; Moore and Arar, 2019). The existing flourishing literature about this aspect (Ahmad et al., 2018; Adamopolou and Moussiades,

2020) has not been associated with equivalent research on methods and linguistic theories that can be pursued for the design phase of conversational projects. During the survey of methodological studies on conversation design, it became clear that there is no shared standard and that various methodological contributions of a practical nature do not refer to a specific theoretical linguistic perspective (Dasgupta, 2018; Pearl, 2016; Cohen et al., 2004; Hall, 2018).

In this work we embrace the Natural Conversation Framework (NCF) whose validity has been already demonstrated in Dall’Acqua and Tamburini (in press); we select some of its most representative patterns and we implemented them on the newly released version of Google Dialogflow CX. This paper is intended as a continuation of the work presented in Dall’Acqua, Tamburini (in press), which sets out the theoretical and methodological assumptions on which this work is based.

2 The Natural Conversation Framework as a Theoretically Funded Approach

Among the linguistic approaches available to analyse interactional exchanges, a *pragmatic perspective* appears to be the most appropriate (Bianchini et al., 2017), especially in its declination of *Conversation Analysis* (Schegloff et al., 1977; Sacks et al., 1974). For this reason, we claim that the *Natural Conversation Framework (NCF)* identified by Moore and Arar (2019), consisting of language patterns structured into sequences in the theoretical groove of Conversation Analysis, could be a promising starting point for the definition of a potentially generalisable and adaptable linguistic methodology.

Since we have already demonstrated the theoretical validity of this approach and we have included it in a practical and applicative procedural workflow on Dialogflow ES (Dall’Acqua and Tamburini, in press) this work aims to continue the

Copyright © 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

research by transposing some of the most significant patterns on the new and very recent version (Nov. 2020) of the platform.

3 Dialogflow CX

The renewed version of Dialogflow is linked to the *information-based* approach (Larsson and Traum, 2000; Traum and Larsson, 2003) and opens to more dynamic scenarios: since it is structured as a finite-state machine, it allows the users to build more flexible, reusable and adaptable patterns. The level of dialogues complexity that can potentially be created is enhanced by the wider range of features that the new tool has to offer: it allows the transition from one state of the conversation to another to be visualized through the creation of *pages*, which are the states of the underlying state machine, configured to collect end-user information relevant to that state of the conversation¹. The conversational flow itself is therefore made of pages, connectors between the pages (known as *state handlers*²) and *flows*, reciprocally independent units of dialogues used to manage more complex conversational agents.

4 Conversational Architecture and Pattern Selection

We enlarged the implementation started in our previous work combining together in an original way a selection of patterns identified by Moore and Arar (2019) and trying to reproduce the most representative, widespread and generalisable use-cases of an high-level conversational agent with practical purposes roughly oriented to customer care. Here, it is not relevant the precise use of the demonstrated chatbot, as the main point is to show and describe the potentialities of the proposed approach. We have adapted patterns taken to all the categories of the classification proposed by Moore and Arar (2019) and suggest that they may also be considered as a best-practice to be taken into account in the summary roadmap towards the implementation previously presented.

In our work we have created three flows. As Fig. 1 shows, they are all connected to the main flow (*Default Start Flow*) both in the initial and in the final state of each flow. The three flows are:

¹<https://cloud.google.com/dialogflow/cx/docs/concept/page>

²<https://cloud.google.com/dialogflow/cx/docs/concept/handler>

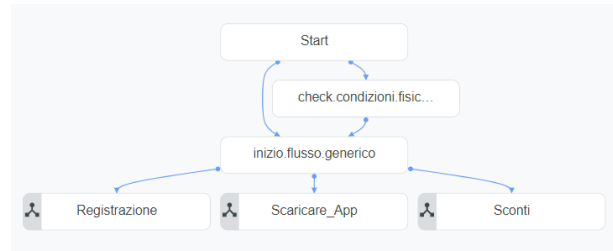


Figure 1: Overall flows architecture.

- **Registration (*Registrazione*):** it reproduces an online registration procedure. It aims to generalise the use-case in which the user has to provide some data (*entities*), divided into mandatory data (without which the procedure cannot succeed) and optional data (the registration can take place correctly even without these data). The procedure of extracting data from the user (*slot filling*) is portable to multiple domains (Mohamad Suhaili et al., 2021) (Fig. 2).

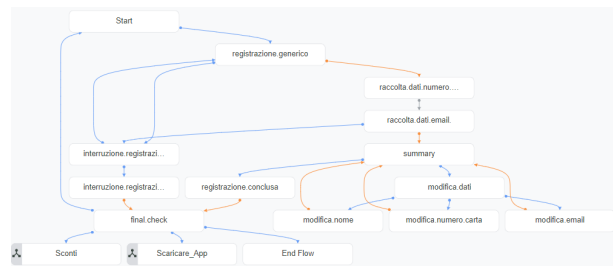


Figure 2: Registration flow diagram.

- **App Download (*Scaricare App*):** it supports the user during a download procedure in multiple steps. It aims to show the application of the *story-telling sequences* (Jefferson, 1978) used to express a content that needs to be parcelled out into smaller pieces of speech. Furthermore, it offers a rudimentary troubleshooting procedure in case of error during the download, that can be actually used to diagnose and manage also other typologies of errors. The widespread of troubleshooting procedures in chatbots is demonstrated in (Thorne, 2017), which also endorses the portability of this type of conversational interactions into multiple domains (Fig. 3).
- **Discounts (*Sconti*):** this flow is dedicated to typologies of discounts and promotions available and it is used to show a combination of patterns that allows to manage series of contextual questions related to the same subject (Fig. 4).

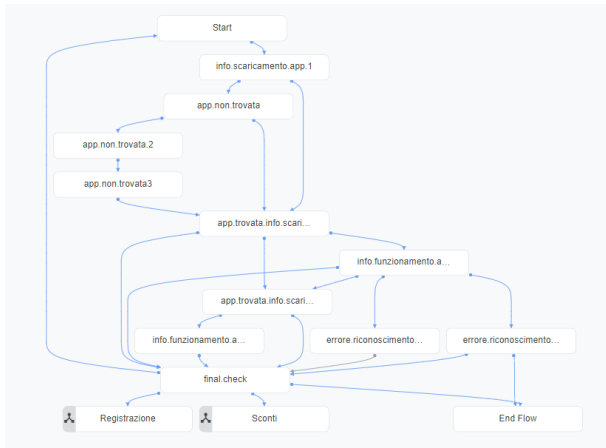


Figure 3: App Download flow diagram.

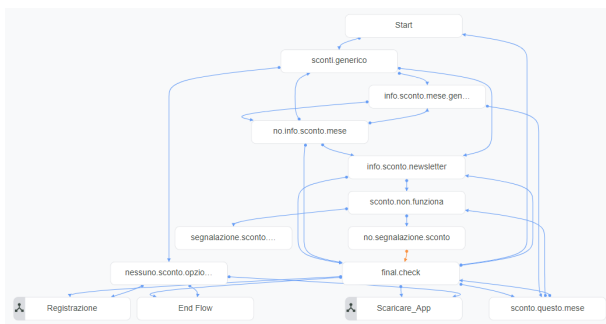


Figure 4: Discounts flow diagram.

Now we are going to review the most pragmatically relevant points of each created flow, aiming to show the adaptability of the patterns also to the Italian language and its suitability to multiple context of use.

4.1 Default Start Flow

The Default Start Flow opens with a slightly revisited version of the pattern **C1.4 Opening Welfare Check (Agent)**³: if the NAME REQUEST in the first turn is for some reasons not satisfied, the agent goes straight to the practical purposes of the conversation, skipping the WELFARE CHECK, as the following dialogues show:

- 1 A: Ciao, sono Teo, il tuo assistente virtuale. Come ti chiami?
- 2 U: Anna, piacere.
- 3 A: Piacere di conoscerti, Anna. Come stai?

³The complete list of all patterns is available in Moore and Arar (2019) and on the website https://researcher.watson.ibm.com/researcher/view_group_subpage.php?id=10182. The Appendix lists the implemented patterns.

- 4 U: al top
Ottimo, sono felice per te! Sono qui per darti qualche informazione su alcuni semplici argomenti:
- 5 A: gli sconti e la procedura di scaricamento della nostra app. Inoltre, posso aiutarti a registrarti sul nostro sito, direttamente in chat. Di che cosa hai bisogno?

- 1 A: Ciao, sono Teo, il tuo assistente virtuale. Come ti chiami?
- 2 U: cavoli miei
Ok, non importa, intanto sai già il mio :) Sono qui per darti qualche informazione su alcuni semplici argomenti: gli sconti e la procedura di scaricamento della nostra app. Inoltre, posso aiutarti a registrarti sul nostro sito, direttamente in chat. Di che cosa hai bisogno?
- 3 A: della nostra app. Inoltre, posso aiutarti a registrarti sul nostro sito, direttamente in chat. Di che cosa hai bisogno?

The agent's reaction in the turn WELFARE RECEIPT (NEUTRAL/POSITIVE/NEGATIVE) is also conditioned by the user's response. From the page **inizio.flusso.generico** (*generic starting flow*) the intents that introduce the flows can be activated, or the user can freely leave the conversation if he/she is not interested in the proposed subjects. In addition, we have also implemented a new version of the pattern **B4.2 Sequence closer Appreciation (helped)**: instead of an admission of not-knowing from the chatbot (NO ANSWER ACCOUNT), we have collocated at one of the initial pages, before that one flow has ever being activated: this condition motivates the answer provided in the following dialog after the thanks from the user:

- 1 A: Ciao, sono Teo, il tuo assistente virtuale. Come ti chiami?
- 2 U: Ciao sono Francesco
- 3 A: Piacere di conoscerti, Francesco. Come stai?
- 4 U: bene
Bene. Sono qui per darti qualche informazione su alcuni semplici argomenti: gli sconti e la procedura
- 5 A: di scaricamento della nostra app. Inoltre, posso aiutarti a registrarti sul nostro sito, direttamente in chat. Di che cosa hai bisogno?
- 6 U: Grazie mille
- 7 A: Non mi pare di aver fatto molto... Tuttavia, prego!

The same pattern is handled differently in the other flows, if the agent actually provided help to the user.

4.2 Registration (*Registrazione*)

When the user directly types something like "I'd like to register to the site" (*Vorrei registrarmi sul sito*), the dedicated flow will be activated and the data extraction procedure will be started. The most appropriate patterns to extract data are: **A2.6 Open Request Summary**, **A2.7 Warrant Request & Refusal**, **A2.11 Open Request Repairs** and **B2.4. Paraphrase Request (elaboration)** combined in original way in order to manage both

mandatory and optional data to complete the on-line registration. Similar patterns have been already implemented in Dialogflow ES (Dall'Acqua and Tamburini, in press), but the output on Dialogflow CX is even more effective also in terms of feasibility of the implementation. As the following dialogue shows

- 1 U: Volevo registrarmi al sito
Per poterti registrare al sito e creare un tuo profilo devo raccogliere qualche tuo dato. Avrei bisogno di nome e cognome, come ad esempio: "Marco Rossi".
- 2 A: Lucia Mondella
- 3 U: Qual è il tuo numero di carta fedeltà?
- 4 A: Quale carta?
- 5 U: La carta fedeltà è una carta che ti permette di raccogliere punti ad ogni tuo acquisto, in negozio e online. Non è obbligatoria, ma attivandola potrai godere di sconti speciali su diversi nostri prodotti. Qual è il tuo numero di carta fedeltà?
- 6 A: Quale numero ti devo dire?
- 7 U: Il codice è a 5 cifre e si trova sul retro della tua carta fedeltà.
- 8 A: non lo vedo
- 9 U: Il codice di carta fedeltà è a cinque cifre (es: "12345") e si trova sul retro della carta, in basso a destra. Qual è il tuo numero di carta fedeltà?
- 10 A: non ce l'ho
- 11 U: Ok, non c'è problema. In ogni caso, potrai aggiornare questa sezione in ogni momento della pagina personale del tuo profilo. Per finire, ho bisogno di un'ultima informazione. Qual è la tua email?
- 12 A: luciamondella@libero.it
- 13 U: Perfetto. Ricapitolando: avvio la procedura di registrazione a nome di Lucia Mondella, numero di carta non fornito, e invio un'email di conferma all'indirizzo luciamondella@libero.it. È tutto corretto?
- 14 A: no, volevo inserire un dato
- 15 U: quale dato volevi inserire?
- 16 A: ho recuperato la carta.
- 17 U: Certamente. Ridigita pure il numero della carta fedeltà. Ti ricordo che è un numero a cinque cifre e si trova sul retro della carta.
- 18 A: 12345
- 19 U: Perfetto. Ricapitolando: avvio la procedura di registrazione a nome di Lucia Mondella, numero di carta 12345, e invio un'email di conferma all'indirizzo luciamondella@libero.it. È tutto corretto?
- 20 A: Ottimo, Lucia Mondella. Abbiamo finito. A breve, riceverai una mail all'indirizzo che mi hai indicato per il completamento della registrazione. Hai bisogno di altre informazioni su sconti, app o registrazione?
- 21 U: no, son a posto
- 22 A: Va bene. Alla prossima!
- 23 U:

we have adapted the selected patterns to the situation, transposing them into questions related to the topic "fidelity card" like "which card?" (*quale carta?*) and "which number should I say?" (*quale numero devo dire?*) and a repair procedure with "I can't see it" (*non lo vedo*). It is noticeable that in case of questions or repairs related to it

that are predictable, the agent reacts accordingly paraphrasing what it has previously said or providing additional information that may help the user to supply the requested data, without stopping to prompt him/her to move the conversation forward ("Which is your fidelity's card number?" *Qual è il numero della tua carta fedeltà?*). The main difference between the same conversational flow on the two versions of Dialogflow is the effectiveness of the final summary, which is a requirement for a correct data extraction procedure (Moore and Arar, 2019). Here the user can choose to not provide an optional data such as the fidelity card number, but if after the final summary he/she chooses instead to insert it, the agent allows the modification and shows the restored value in the next REQUEST SUMMARY, asking again if the summary is correct or not. If is it correct, this flow is going to an end and it proposes one of the last pages of it, namely the **final.check**, realised with the pattern **C4.0 Last Topic Check (Agent)** in which the agent invites the user or to transit from another flow, or to freely interrupt the conversation. Another relevant ramification of the flow "Registrazione" shows the management of an obligatory data, such as the name:

- 1 A: Mi devo registrare al sito
Per poterti registrare al sito e creare un tuo profilo devo raccogliere qualche tuo dato. Avrei bisogno di nome e cognome, come ad esempio: "Marco Rossi".
- 2 U: perché?
- 3 U: Il nome è un requisito essenziale per la registrazione. Non preoccuparti, sarà gestito come dato personale e sarà visibile solo a te. Come ti chiami?
- 4 A: trazione. Non preoccuparti, sarà gestito come dato personale e sarà visibile solo a te. Come ti chiami?
- 5 U: non voglio dire il mio nome ad un robot
- 6 A: Capisco. Senza questo dato, però, non posso aiutarti con la registrazione. Preferisci interrompere qui?
- 7 U: no va bene
- 8 A: Va bene. Riprendiamo allora da dove eravamo rimasti. Per poterti registrare al sito e creare un tuo profilo devo raccogliere qualche tuo dato. Avrei bisogno di nome e cognome, come ad esempio: "Marco Rossi".
- 9 U: Stupido robot
- 10 A: Non serve a nulla reagire in modo aggressivo. Come ti chiami?
- 11 U: Cretino!
- 12 A: In questo modo stai ferendo il mio povero cuore metallico. Come ti chiami?
- 13 U: non te lo dico
- 14 A: Capisco. Senza questo dato, però, non posso aiutarti con la registrazione. Preferisci interrompere qui?
- 15 U: sì ciao

Also in this case we have provided a series of specific question related to the topic such as "why?" (*perché?*), but if the user refuses to provide this data, the chatbot suggests to interrupt the proce-

ture. Since the goal of this demo is to show the potentialities of this methodology, we have stressed the possibilities of the conversation making the user going back to the previous procedure, apparently willing to provide the name. But once the data extraction procedure starts again, the user starts to insult the robot, that reacts accordingly to every insult but without stopping to ask for the name, until the user communicates his/her will of interrupting the procedure and the conversation.

4.3 App Download (*Scaricare App*)

We have figured this flow out to show the transponibility of patterns **A3.0 Extended Telling with Repair** and **A3.1 Extended Telling Abort**, that are normally used to expose a story, an anecdote or a list of instructions that cannot be achieved in one single conversational turn. This flow can be activated taping an expression associated to the corresponding intent, such as “how do I download the app?” (*come scarico l'app?*), that leads to the first step of the explanation, with an extra-prompt to the user (“Did you find it?” *L’hai trovata?*) to move the conversation forward. If the user claims he/she did not find it, the agent provides additional information to make the recognition easier, which may eventually end up in troubleshooting procedure carried out by the pages “**app.non.trovata**”, “**app.non.trovata.2**”, “**app.non.trovata.3**”.

- 1 U: come si scarica l'app?
Prima di tutto, vai su App Store o su Playstore
- 2 A: e digita il nome della nostra catena nella barra di ricerca. Così facendo dovresti vedere la nostra app. L’hai trovata?
- 3 U: Non la vedo
L’app si chiama esattamente come la nostra catena di negozi ed è rappresentata da un triangolo di colore verde. Prova a uscire rientrare dall’App Store che stai utilizzando e a digitare nuovamente il nome nella barra di ricerca. Adesso l’hai trovata?
- 4 A: eccola
- 5 U: Perfetto, adesso puoi scaricarla. Ci vorrà solo qualche secondo.
- 6 A: come si fa?
- 7 U: Facendo tap su “installa” dovresti riuscire a scaricare l’app sul tuo smartphone.
- 8 A: ok
- 9 U: A questo punto, una volta installata, ti basta inserire la tua mail e il tuo numero della carta fedeltà per associare la tua app al tuo profilo online, aggiornare tutte le tue informazioni e poter fare acquisti.
- 10 A: grazie mille
- 11 U: Prego! Hai bisogno di altre informazioni su sconti, app o registrazione?
- 12 A:

The last two pages represent the final states of troubleshooting, which therefore may not be in-

voked if the user has solved the problem in the previous states.

The conversation is articulated in a sequence of instructions: each following step in the explanation is motivated by the `CONTINUER` tipped by the user, that allows the agent to provide another piece of information until the procedure is finished (Schegloff, 1982). The conversational turns in which the user simply reports that the previous step has been received can be alternated with repair procedures, in which the user asks for explanations regarding a specific step (Sacks et al., 1992), as shown in the dialogue above. In every moment of the procedure, the user may also notify that he/she has actually completed the procedure through an intent categorised as `COMPLAINT/ABORT REQUEST` and the agent should therefore stop providing instructions. In our flow the agent acknowledges the user’s request (“Great!” *Ottimo!*) and steps forward to the page **final.check**:

- 1 U: come faccio a scaricare l’app?
Prima di tutto, vai su App Store o su Playstore e digita il nome della nostra catena nella barra di ricerca. Così facendo dovresti vedere la nostra app. L’hai trovata?
- 2 A: sì
- 3 U: Perfetto, adesso puoi scaricarla. Ci vorrà solo qualche secondo.
- 4 A: Ok basta ho risolto grazie
- 5 U: Ottimo! Hai bisogno di altro su sconti, app oppure registrazione?
- 6 A:

4.4 Discounts (*Sconti*)

With this last flow we would like to demonstrate the applicability of two more patterns, namely the pattern **A2.9 Open Request Incremental** **A2.10 Open Request Series**, **A2.4 Open Request Agent Detail Request**. They are used to manage situations in which the users asks a series of questions related to a single topic or switches topic but asks similar questions related to it. As the following dialogue shows

- 1 U: Che sconti avete?
Esistono due principali tipologie di sconto: lo
- 2 A: sconto newsletter e lo sconto del mese. Su quale di queste vorresti ricevere maggiori informazioni?
- 3 U: il primo
Iscrivendoti alla nostra newsletter riceverai uno
- 4 A: sconto di del 10% utilizzabile una sola volta su una spesa di almeno 30 euro, online o in negozio.
- 5 U: Posso usarlo anche per un acquisto online?
- 6 A: Non ho capito, puoi ripetere la tua domanda sullo sconto newsletter?
- 7 U: posso usare lo sconto newsletter sia online sia in negozio?

- 8 A: Puoi utilizzare lo sconto newsletter su tutti i tuoi acquisti, sia online, sia in negozio.
- 9 U: Come funziona?
Basta cliccare su “Iscrizione newsletter” in alto a destra in ogni pagina del sito e inserire la tua email nella tab che si apre. Presto arriverà in posta un codice sconto a sei cifre che potrai applicare ad un acquisto online o in negozio a partire da 30 euro di spesa.
- 10 A: ok
- 11 U: Hai bisogno di altro sugli sconti, sulla procedura di registrazione, o su come scaricare l’app?
- 12 A: volevo delle informazioni sullo sconto del mese
Per il mese di gennaio, prevediamo una promozione sugli articoli per bambini, con uno sconto del 50% su ogni prodotto indicato con il simbolo “Child” sia online sia in negozio.
- 13 U: Quando scade?
- 14 A: Questo tipo di sconto si rinnova di mese in mese:
questo, ad esempio, sarà valido fino al 31 di gennaio.
- 15 U: come si usa?
- 16 A: Lo sconto verrà automaticamente applicato se il prodotto scelto per l’acquisto è oggetto della promozione: se stai acquistando online, verrà applicato il prezzo scontato indicato accanto al protocollo, altrimenti ti verrà applicato lo sconto direttamente in cassa.
- 17 U: ok grazie mille
- 18 A: Prego! Hai bisogno di altro sugli sconti, sulla procedura di registrazione, o su come scaricare l’app?
- 19 U: no grazie
- 20 A: Va bene. Alla prossima!

the conversation opens with a generic open request from the user (“what kinds of discounts do you have?” *che sconti avete?*) that is satisfied by the agent with a request of more details: the agent needs to know the specific type of discount as additional detail to provide specific information about it. Once the type of discount is defined, the user can start asking specific question related to it without always specifying the subject. Once the user has satisfied his/her needs in relation to newsletter discount, at the **final.check** page he/she can switch the topic and start asking a series of question related to the other one, until the user has achieved all the needed information. Once the user thanks the agent, this is perceived as an acknowledgement of a successful conversation, so the conversational flow can go away.

5 Conclusions and Future Directions of the Research

We have demonstrated the applicability of this method also on the new released version of one of the most important Natural Language Understanding platform, namely Dialogflow CX. Since this version of Dialogflow has been released for the Italian language only in November 2020, to

our knowledge this is the only study in which this platform is used for Italian and for the realisation of a conversational project with practical purposes. This is therefore a further and more complete implementation of the pragmatic adequateness of this approach.

In the future, we would like to evaluate the effectiveness of this approach on a fully functional prototype that can be adapted not only for commercial purposes, but also for other important application contexts, such as education. Due to the variety of frameworks available for the evaluation process (Casas et al., 2020), the question of the most suitable evaluation method is still open for further discussion.

Acknowledgements

We would like to thank Injenia S.r.l. for supporting this research. CRediT author statement; ADA: Conceptualization, Methodology, Validation, Formal Analysis, Investigation, Writing (Original Draft), Writing (Review & Editing); FT: Conceptualization, Supervision, Project Administration, Writing (Review & Editing), Funding Acquisition.

References

- E. Adamopolou and L. Moussiades. 2020. An overview of chatbot technologies. *Artificial Intelligence Applications and Innovations*, 584:373–383.
- N. A. Ahmad, M. H. Che, A. Zainal, M. A. R. Fairuz, and Z. Adnan. 2018. Review of chatbots design techniques. *International Journal of Computer Applications*, 181(8):7–10.
- A. Bianchini, F. Tarasconi, R. Ventaglio, and M. Guadalupi. 2017. “gimme the usual” - how handling of pragmatics improves chatbots. In *Proceedings of the Fourth Italian Conference on Computational Linguistics (CLiC-it 2017)*, pages 30–35.
- D. Braun and F. Matthes. 2019. Towards a framework for classifying chatbots. In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 484–489. volume 1.
- J. Casas, M.-O. Tricot, O. Abou Khaled, E. Mugellini, and P. Cudré-Mauroux. 2020. Trends & methods in chatbot evaluation. In *Companion Publication of the 2020 International Conference on Multimodal Interaction*, page 280–286, New York, NY, USA. Association for Computing Machinery.
- A. P. Chaves, E. Doerry, J. Egbert, and M. Gerosa. 2019. It’s how you say it: Identifying appropriate register for chatbot language design. In *Proceedings of the 7th International Conference on Human-Agent Interaction*, pages 102–109.

- M. H. Cohen, J. P. Giangola, and J. Balogh. 2004. *Voice User Interface Design*. Addison Wesley.
- K. M. Colby, S. Weber, and F. D. Hilf. 1971. Artificial paranoia. *Artificial Intelligence*, 2(1):1–25.
- R. Dale. 2016. Industry watch the return of chatbots. *Natural Language Engineering*, 22(5):811–817.
- A. Dall’Acqua and F. Tamburini. in press. Toward a linguistically grounded dialog model for chatbot design. *Italian Journal of Computational Linguistics*, 7(1).
- R. Dasgupta. 2018. *Voice User Interface Design. Moving from GUI to Mixed Modal Interaction*. Apress.
- A. Følstad, M. Skjuve, and P. B. Brandtzaeg. 2019. Different chatbots for different purposes: Towards a typology of chatbots to understand interaction design. In S. S. Bodrunova, O. Koltsova, A. Følstad, H. Halpin, P. Kolozaridi, L. Yuldashev, A. Smoliarova, and H. Niedermayer, editors, *Internet Science. INSCI 2018. Lecture Notes in Computer Science*, pages 145–156. vol 11551, Springer, Cham.
- E. Hall. 2018. *Conversational design*. A Book Apart.
- S. Hussain, O. Ameri Sianaki, and N. Ababneh. 2019. A survey on conversational agents/chatbots classification and design techniques. In L. Barolli, M. Takizawa, F. Xhafa, and T. Enokido, editors, *Web, Artificial Intelligence and Network Applications. WAINA 2019. Advances in Intelligent Systems and Computing*. vol 927. Springer, Cham.
- G. Jefferson. 1978. Sequential aspects of storytelling in conversation. In J. Schenkein, editor, *Studies in the Organization of Conversational Interaction*, pages 219–48. Academic Press.
- S. Larsson and D.R. Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural Language Engineering*, 6(3-4):323–340.
- V. Mathur and A. Sing. 2018. The rapidly changing landscape of conversational agents. *arXiv 1803.08419v2*, pages 1–14.
- S. Mohamad Suhaili, N. Salim, and M. Nazim Jambli. 2021. Service chatbots: A systematic review. *Expert Systems with Applications*, 184:115461.
- R. J. Moore and R. Arar. 2019. *Conversational UX Design. A Practitioner’s Guide to the Natural Conversation Framework*. Association for Computing Machinery.
- C. Pearl. 2016. *Designing Voice User Interfaces. Principles of Conversational Experiences*. O’Reilly.
- N. Radziwill and M. Benton. 2017. Evaluating quality of chatbots and intelligent conversational agents. *arXiv 1704.04579*.
- H. Sacks, E. A. Schegloff, and G. Jefferson. 1974. A simplest systematics for organization of turn-taking for conversation. *Language*, 50:696–735.
- H. Sacks, E. Schegloff, and G. Jefferson. 1992. *Lectures on Conversation*, volume I, II. Blackwell.
- E. A. Schegloff, G. Jefferson, and H. Sacks. 1977. The preference for self-correction in the organization of repair in conversation. *Language*, 53(2):361–382.
- E.A. Schegloff. 1982. Discourse as an interactional achievement: Some uses of ‘uh huh’ and other things that come between sentences. In D. Tannen, editor, *Analysing Discourse: Text and Talk*, page 71–93. Washington DC, Georgetown University Press.
- C. Thorne. 2017. Chatbots for troubleshooting: A survey. *Language and Linguistics Compass*, 11(10):e12253. e12253 LNCO-0688.R1.
- D.R. Traum and S. Larsson. 2003. The information state approach to dialogue management. In J. van Kuppevelt and R.W. Smith, editors, *Current and New Directions in Discourse and Dialogue*, pages 325–353. Springer Netherlands, Dordrecht.
- M. Tsvetkova, R. García-Gavilanes, L. Floridi, and T. Yasseri. 2017. Even good bots fight: The case of wikipedia. *PLOS ONE*, 12(2):1–27.
- J. Weizenbaum. 1966. Eliza – a computer program for the study of natural language communication between man and machine. *CACM*, 9(1):36–45.

Appendix

List of the implemented patterns:

C1.4 Opening Welfare Check (Agent)

- 1 A: GREETING. SELF-IDENTIFICATION. NAME REQUEST.
- 2 U: NAME.
- 3 A: GREETING, DIRECT ADDRESS. WELFARE CHECK.
- 4 U: WELFARE REPORT (NEUTRAL / POSITIVE/ NEGATIVE)
- 5 A: WELFARE RECEIPT (NEUTRAL / POSITIVE / NEGATIVE)

B4.3 Seq. Closer Appreciation (not helped)

- 1 A: INQUIRY/REQUEST
- 2 U: NO ANSWER ACCOUNT
- 3 A: APPRECIATION
- 4 U: REFUTATION

A2.6 Open Request Summary

- 1 U: PARTIAL REQUEST
- 2 A: DETAIL REQUEST
- 3 U: DETAIL
- 4 A: DETAIL REQUEST
- 5 U: DETAIL
- 6 A: DETAIL REQUEST
- 7 U: DETAIL
- 8 A: REQUEST SUMMARY
- 9 U: SUMMARY CONFIRM
- 10 A: GRANT
- 11 U: SEQUENCE CLOSER
- 12 A: RECEIPT

A2.7 Warrant Request & Refusal

- 1 A: DETAIL REQUEST
- 2 U: WARRANT REQUEST
- 3 A: WARRANT
- 4 U: REFUSAL
- 5 A: ACKNOWLEDGEMENT. <NEXT SECTION>

A2.11 Open Request Repairs

- 1 U: FULL REQUEST
- 2 A: GRANT
- 3 U: REPAIR INITIATOR
- 4 A: REPAIR
- 5 U: SEQUENCE CLOSER
- 6 A: RECEIPT

B2.4 Paraphrase Request (elaboration)

- 1 U: <ANY UTTERANCE>
- 2 A: PARAPHRASE REQUEST
- 3 U: PARAPHRASE DEFAULT

A3.0 Extended Telling with Repair

- 1 A: STORY/INSTRUCTION INVITATION
- 2 U: PART/STEP 1
- 3 A: CONTINUER/PAUSE
- 4 U: PART/STEP 2
- 5 A: REPAIR INITIATOR
- 6 U: REPAIR
- 7 A: CONTINUER/PAUSE
- 8 U: PART/STEP 3
- 9 A: SEQUENCE CLOSER
- 10 U: RECEIPT

A3.1 Extended Telling Abort

- 1 A: STORY/INSTRUCTION INVITATION
- 2 U: PART/STEP 1
- 3 A: CONTINUER/PAUSE
- 4 U: PART/STEP 2
- 5 A: REPAIR INITIATOR
- 6 U: REPAIR
- 7 A: PART/STEP 3
- 8 U: COMPLAINT/ABORT REQUEST
- 9 A: ABORT OFFER
- 10 U: ABORT CONFIRM
- 11 A: ACKNOWLEDGEMENT

A2.9 Open Request Incremental

- 1 U: FULL REQUEST
- 2 A: GRANT
- 3 U: INCREMENTAL REQUEST
- 4 A: GRANT
- 5 U: SEQUENCE CLOSER
- 6 A: RECEIPT

A2.10 Open Request Series

- 1 U: FULL REQUEST
- 2 A: GRANT
- 3 U: RELATED REQUEST
- 4 A: GRANT
- 5 U: SEQUENCE CLOSER
- 6 A: RECEIPT

A2.4 Open Request Agent Detail Request

- 1 U: PARTIAL REQUEST
- 2 A: DETAIL REQUEST
- 3 U: DETAIL
- 4 A: GRANT
- 5 U: SEQUENCE CLOSER
- 6 A: RECEIPT