UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Francisco Albernaz Machado Valério

Chatbots and Communication:
A Qualitative Analysis of Communicative Aspects of Conversational Interfaces

Belo Horizonte
2020

Francisco Albernaz Machado Valério

Chatbots and Communication:
A Qualitative Analysis of Communicative Aspects of Conversational Interfaces

Final Version

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Raquel Oliveira Prates
Co-Advisor: Heloisa Caroline de Souza Pereira Candello

Belo Horizonte
2020

Valério, Francisco Albernaz Machado

V164c Chatbots and communication [recurso eletrônico]: a qualitative analysis of communicative aspects of conversational interfaces / Francisco Albernaz Machado Valério – 2020.
1 recurso online (173 f. il, color.): pdf.

Orientadora: Raquel Oliveira Prates.
Coorientadora: Heloisa Caroline de Souza Pereira Candello
Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 163-168.

1. Computação – Teses. 2. Chatbots – Teses. 3. Engenharia Semiótica – Teses. 4. Método de Inspeção Semiótica-– Teses. I. Prates, Raquel Oliveira. II. Candello, Heloisa Caroline de Souza Pereira III. Universidade Federal de Minas Gerais; Instituto de Ciências Exatas Departamento de Ciência da Computação. IV. Título.

CDU 519.6*75(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa
CRB 6/1510 Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Chatbots and Communication: A Qualitative Analysis of Communicative Aspects of Conversational Interfaces

FRANCISCO ALBERNAZ MACHADO VALÉRIO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Profa. Raquel Oliveira Prates - Orientadora
Departamento de Ciência da Computação - UFMG

Dra. Heloisa Caroline de Souza Pereira Candello - Coorientadora
IBM Research - IBM Brasil

Prof. Cristiano Maciel
Instituto de Computação - UFMT

Dr. Thiago Castro Ferreira
Pós-Doutorando - Faculdade de Letras - UFMG

Prof. Pedro Olmo Stancioli Vaz de Melo
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 2 de Julho de 2020.

Acknowledgments

I would like to acknowledge everyone that contributed to this work. Thanks to both my advisors Raquel Prates and Heloisa Candello, all members of the dissertation committee, everybody that participated in the study, and Thiago D'Ávila, for letting me use his chatbot for user tests. I also would like to thank the Graduate Program in Computer Science of the Federal University of Minas Gerais and National Council for Scientific and Technological Development for providing the material means for this work.

Finally, special thanks to Tati, that helped me in the first and hardest steps of this work, coauthored the paper that originated this dissertation, and convinced me to pursue my master's in the HCI field.

“*Unless in communicating with a computer one says exactly what one means, trouble is bound to result.*”

(Alan Turing)

Resumo

Recentemente, a popularidade de chatbots baseados em texto aumentou, possivelmente devido a novas APIs de redes sociais e serviços de mensagens e plataformas de desenvolvimento que ajudam a lidar com o Processamento de Linguagem Natural necessário. Entretanto, como os chatbots usam principalmente linguagem natural como interface, usuários podem ter problemas para descobrir quais frases os chatbots vão conseguir entender e o que eles conseguem fazer. Por isso é importante apoiar projetistas na hora de decidir como transmitir aos usuários quais são as funcionalidades do chatbot, uma vez que isso pode determinar se o usuário vai continuar utilizando-o ou não. O objetivo deste trabalho é analisar as estratégias de comunicação utilizadas em chatbots populares para mostrar suas funcionalidades a usuários. Utilizando o Método de Inspeção Semiótica (MIS), descobrimos 11 estratégias para informar funcionalidades a usuários utilizadas nos chatbots analisados. Então analisamos outros dez chatbots para consolidar essas descobertas. Na sequência, realizamos uma pesquisa exploratória utilizando aspectos pragmáticos como Atos de Fala e Princípios Cooperativo e da Polidez para analisar a comunicação com chatbots. Também foram conduzidos estudos com usuários comparando abordagens distintas no projeto de chatbots e o uso das estratégias identificadas anteriormente. Finalmente, discutimos o uso dessas estratégias, os desafios no projeto desse tipo de interface, limitações do uso do MIS e da nossa metodologia neste contexto, e a percepção de usuários sobre diferentes maneiras de interação com chatbots e suas estratégias de comunicação.

Palavras-chave: interação humano-computador, interfaces conversacionais, engenharia semiótica

Abstract

Recently, text-based chatbots had a rise in popularity, possibly due to new APIs for online social networks and messenger services, and development platforms that help to deal with all the necessary Natural Language Processing. However, as chatbots mainly use natural language as interface, users may struggle to discover which sentences the chatbots will understand and what they can do. So it is important to support designers in deciding how to convey the chatbots' features, as this might determine whether the user will continue chatting or not. In this work, our goal is to analyze the communicative strategies used by popular chatbots when conveying their features to users. We used the Semiotic Inspection Method (SIM) for that end, and we were able to identify 11 strategies used by the analyzed chatbots for conveying their features to users. We then consolidated these findings by analyzing another ten chatbots. Later, we further analyzed the chatbot communication with an exploratory investigation of Pragmatics aspects of Speech Acts and Cooperative and Politeness Principles. We also conducted user studies comparing different approaches to chatbot design and the use of the previously identified strategies. Finally, we discuss the use of these strategies, challenges for designing such interfaces, limitations of using SIM and our methodology on them, and users' perception of different ways of interacting with chatbots and their communicative strategies.

Keywords: human-computer interaction, conversational interfaces, semiotic engineering

Contents

Chapter 1

Introduction

Command-line interfaces are a common way of interacting with computers. They evolved from teletypes, and, in the 1970s, the widespread adoption of video-display terminals made them the primary way of interacting with computer systems. Users would type specific terms from a specialized vocabulary that could be interpreted by the system. After the command was issued, the system would process it and print an answer to the user, similar to a conversation (Raymond and Landley, 2004).

In time, graphical user interfaces came around, as well as the mouse, which allowed users to interact through clicking and manipulating elements on the screen. That style of interface became very popular, to the point of being considered the standard way of interacting with computers. Recently, due to the popularization of smartphones and tablets, touch-based interfaces allowed for interaction without mice or physical keyboards.

Nonetheless, since the sixties we have also been able to talk to computers through intelligent software such as STUDENT (Bobrow, 1964) and ELIZA (Weizenbaum, 1966), that allow for interaction using text-based natural language as input and produces a natural language output (Candello and Pinhanez, 2016).

Since then, chatbot technology has greatly advanced. During the early 2010s, some *Intelligent Personal Assistants* were made available to the public, specially *Siri*[1], *Google Assistant*[2], *Cortana*[3], and *Alexa*[4]. These assistants can help users with many types of tasks, such as sending messages, adding events to calendars, searching the web, among others. The paramount way of interaction with these assistants is through natural language, either speech or text-based.

Chatbot APIs for Online Social Networks and messenger applications (e.g. Facebook Messenger and Telegram), and toolkits that help developers incorporate AI, Machine Learning, and Natural Language Processing (NLP) in an easy-to-learn way, with little to no programming needed (such as IBM's Watson Assistant Service, Wit.ai, Amazon Lex, and Microsoft's LUIS, among others) helped to advance chatbot technology in recent years (McTear, 2017).

---

[1] `https://www.apple.com/siri/`, last access on Jun/2020.
[2] `https://assistant.google.com/`, last access on Jun/2020.
[3] `https://www.microsoft.com/en-us/cortana/`, last access on Jun/2020.
[4] `https://developer.amazon.com/alexa`, last access on Jun/2020.

Maintaining context in a conversation and allowing natural language interactions are among the characteristics of modern conversational interfaces (Candello and Pinhanez, 2016). Besides the Intelligent Personal Assistants, conversational interfaces are also used on chat apps, like the late *Google Allo*[5], discontinued in March 2019, *Telegram*[6], and *Facebook Messenger*[7] to provide services and value to their users. Some examples include *KLM*'s chatbot that allows users to check information about their flights and can also be used to change seats. Another example of a modern conversational interface is *Slack*[8]'s *Slackbot*, which helps Slack users in tasks such as scheduling meetings with other users.

The market for conversational interfaces is still growing (VB Profiles, 2018). In 2017, predictions said that the chatbot market size would reach US$11,25 billion by 2025 (Grand View Research, 2017). Gartner predicted that by 2021 most companies would spend more on chatbots than on traditional mobile apps (Gartner, 2018). In 2018, however, some predicted the doomsday of chatbots when Facebook's personal assistant M was discontinued (Simonite and Griffith, 2018). In 2019, predictions point to a thriving market accounted for US$1,072 billion in 2018 and is expected to reach US$ 9,475 billion by 2027 (Research and Markets, 2019).

Chatbots are different from traditional graphical user interfaces, as they use text or speech-based natural language (Candello and Pinhanez, 2016) instead of other graphical elements such as buttons and menu bars other systems use. In traditional graphical user interfaces, designers make use of signs[9] from shared signification systems to guide users through the interface. Many visual cues can be used to convey to users the expected interaction. For example, checkboxes to choose multiple relevant options or radio buttons for mutually exclusive alternatives. However, when designing chatbots there are fewer cues and affordances to choose from. This lack of good non-visual affordances makes it difficult to design good conversational interfaces (Candello and Pinhanez, 2016). Chatbots unveil themselves to the user one sentence at a time. Because of that, users may struggle to interact with them and to understand what they can do. Hence, it is important to support designers in deciding how to convey chatbots' features to users, as this might determine whether the user continues to chat or not.

Although there are works focusing on evaluation methods to assess text-based chatbots, these focus on specific components (McTear et al., 2016b), such as benchmarks for Natural Language Processing and annotated results for queries made to the chatbot (Jiang et al., 2015), with only a few qualitative evaluation methods for text-based chatbots evaluation (McTear et al., 2016b). Little is known about the best strategies to

[5]`https://allo.google.com/`, last access on Jun/2020.
[6]`https://telegram.org/`, last access on Jun/2020.
[7]`https://www.messenger.com/`, last access on Jun/2020.
[8]`https://slack.com/`, last access on Jun/2020.
[9]In the present work we use the term "sign" in the Peircian sense, of anything that means something to someone (Peirce, 1992).

use when conveying chatbots' features to users. Some works close to this subject explores how distinct typefaces can influence users' perception of whether they are talking to a person or a bot (Candello et al., 2017); and a study with first-time users shows aspects they found most important when interacting with chatbots (Jain et al., 2018).

The initial conversation with a chatbot is often accompanied by a sensation of "what can I say to you?". That sensation usually goes away once you learn what the chatbot can do and how to do it. On the other hand, if it takes too long to discover what the chatbot can do, you are probably going to abandon the conversation. With that in mind, we decided to explore how chatbots make users know what they can do.

1.1 Objectives

In this work, our main goal is to identify and explore communicative strategies used by popular chatbots to convey their features to users. To do so, we conducted an exploratory study aimed at identifying what were the strategies being used in existing successful chatbots, and organizing them so other designers can benefit from them to make better chatbots.

Our study is divided into four parts.

In the first part, we performed systematic analyses of existing popular chatbots with the goal of identifying which strategies (if any) they used to communicate their features to users. As result, we identified some communicative strategies used by designers to convey the chatbots' features to users, as well as classes of visual cues used on the chatbots. This study was first published on (Valério et al., 2017).

Next, we sought to consolidate our findings. To do so, we adopted a top-down approach, using our previous findings to guide a second round of inspections. This second part comprehended a larger set of ten chatbots, enabling us to have a broader view and to consolidate our previous findings. This part of the work was first published on (Valério et al., 2018).

Subsequently, we explored these strategies and sign classes using some Pragmatics theories to better understand the communication that was taking place through the chatbots. By doing so, we gained new insights about this communication, as well as on the strategies and sign classes.

Finally, we created a chatbot called *Cinemito*, that implemented the same features of an existing chatbot (*Kino*) using a different design approach. That allowed us to analyze users' perception of some of our findings from previous results by conducting interviews and user experiments with two chatbots with different interaction styles and strategies. The user research yielded results that also supported those previous findings.

1.2 Contributions

As result, we were able to consolidate a list of sign classes used to assemble the chatbot's interface and communicative strategies for conveying chatbots' features to users that emerged from the inspections. Furthermore, one more strategy was identified in the second round and added to the set of strategies. It is important to note that the sign classes and strategies have distinct objectives, as the sign classes form a framework for the chatbot design, while the strategies focus on achieving particular objectives (conveying features, in this case). These results can support chatbot designers (as well as chatbot programmers and domain experts) when deciding how and when to communicate chatbots' features to users.

The discussion regarding the use of Pragmatics elements (Speech Acts and Cooperative and Politeness Principles) to assess chatbot interaction, although tested in just a few chatbots, was useful for identifying and explaining how and why some communication breakdowns took place. Our analysis indicates that chatbot research and design could benefit from analyses guided by these elements.

Our user study compared two chatbots with the same features but different approaches to their design. It depicts and allows for a view of the user experience concerning the use of sign classes and strategies. Moreover, it also raises interesting points about users' expectations when interacting with different types of chatbots, as well as their perception of intelligence for that area.

We also discuss two main challenges for designers: the openness of the communication space and the hidden structure that are part of the chatbots' essence. Some recommendations for dealing with that are also discussed.

Finally, this work also brings methodological contributions regarding the use of SIM and the maxims of the Cooperative and Politeness Principles on conversational interfaces, which can prove useful for further research on both conversational interfaces and Semiotic Engineering.

1.3 Overview

This work is organized as follows. First, we present related work on chatbots in Chapter 2 - “Related works”. Next, in Chapter 3 - “Methodology” we show an overview of the methodology used in this work, as the details of it are included in the remaining chapters.

The semiotic inspections, including detailed methodology and premises for applying SIM to conversational interfaces, as well as the sign classes and strategies for conveying features resulting from the two rounds of inspections are shown in Chapter 4 - “Communicative Strategies Investigation”. This chapter also includes a discussion about certain characteristics particular to conversational interfaces, such as the openness of their interaction space and possible ways for dealing with it.

Chapter 5 - “Pragmatics and Chatbots” is where we show the Pragmatics theories used to further analyze the chatbots’ sentences, namely Speech Acts, Cooperative Principle, and Politeness Principle, as well as the premises for using them to analyze sentences uttered on conversational interfaces. We then show a few classification examples of chatbots’ sentences, to better illustrate the methodology of that step of our work. The insights from the analysis and the relation of these Pragmatics theories to our previous findings are also discussed.

Later, in Chapter 6 - “User tests”, we show the user tests and interviews we conducted with two chatbots to analyze their perception of the strategies, sign classes, and interaction styles. Then we discuss how some of those emerged during interviews.

Finally, Chapter 7 - “Final Remarks” contains this work’s final remarks, conclusions, and contributions.

Chapter 2

Related works

This chapter presents a brief history of conversational interfaces, the state of the art of chatbots, and relevant work on text-based chatbot design and interaction aspects.

2.1 Conversational Interfaces Through Time

This section shows a very brief set of examples of conversational systems throughout history, with few examples ranging from usual software to video games.

During the 1960s the first conversational interfaces were created. One example is *STUDENT* (Bobrow, 1964), a chatbot that helped students solve high-school algebra problems using natural language.

In the same decade, *ELIZA* was created (Weizenbaum, 1966). *ELIZA* simulated a Rogerian psychotherapist and allowed users to communicate using natural language input. It would then search for keywords in users' input for identifying context and reply an answer, often rephrasing a previous sentence as a question, for example, if the user said "I'm mad at my father", ELIZA would respond "Why are you mad at your father?", like a psychiatrist.

From 1968 to 1970, Terry Winograd worked on *SHRDLU*[1], which was able to understand basic English statements and commands. Users could tell the system how to manipulate objects, and *SHRDLU* would ask for clarifications if it could not understand the sentence (Winograd, 1971 apud Candello and Pinhanez, 2016).

More recently, in 1995, the development of *A.L.I.C.E.* [2] (Artificial Linguistic Internet Computer Entity) began (Thompson, 2002). Created by Richard Wallace and inspired by *ELIZA*, *ALICE* is open-source, distributed under *GNU General Public License*, and uses AIML (Artificial Intelligence Markup Language) to create responses to inputs.

[1] `http://hci.stanford.edu/~winograd/shrdlu/`, last access on Jun/2020.

[2] `http://alicebot.sourceforge.net/alice_page.htm`, last access on Jun/2020.

In 1997, the world met the infamous *Clippit* [3], commonly known as *Clippy*, Microsoft Office's assistant created by Kevan J Attebery. Many users found *Clippy*'s frequent interruptions annoying, and it was later discontinued (Candello and Pinhanez, 2016).

The video game *Seaman*[4] was published by *Sega* in 1999 for the *Dreamcast* game console. In this game, players had to raise a fish-like creature with a human face called *Seaman*, to do that, players need to feed it and also talk to it through a microphone. During some moments, the creature would ask questions about the player's life and opinions and comment on their answers (Tieryas, 2017).

The conversational interfaces above were a few examples of software using natural language input (either through text or voice) before the modern chatbots and intelligent personal assistants that were popularized after the release of *Siri*, *Google Assistant*, and *Alexa*.

The first modern intelligent personal assistant to reach the market was Apple's *Siri*, released in 2011 for *iOS* devices (McTear et al., 2016a). As it was released for a very popular platform, at its launch, *Siri* already had millions of potential users (Luger and Sellen, 2016). *Siri* is activated through voice commands and it is able to use some of the smartphone features, such as making phone calls, turning on the flashlight, or sending messages. It can also answer some questions, having some predetermined replies to some small talk interactions (such as "hello", or "how are you?"), and searching the web for other questions. Being an assistant with its own voice and some personality traits, and responding to questions in natural language, some users even treat *Siri* as a real person, asking "please" and thanking it after interactions (Leahu et al., 2013).

Released in 2012, *Google Now*[5] was Google's first attempt at creating a personal assistant. Initially, it was available only to *Android* smartphones, but later some of it was also offered on other platforms, such as *iOS* devices, *Google Home* smart speakers, and web browsers. Like *Siri*, *Google Now* was also voice-activated. It also could control some functions of the smartphone and compatible smart home appliances on *Google Home*, and respond to questions by searching the web with Google. On smartphones, it also preemptively showed relevant information based on users' habits through cards (Candello and Pinhanez, 2016).

Later, in 2016, Google released *Google Assistant*[6], its second personal assistant, which inherited *Google Now*'s features but also allowed for two-way communication through a chat interface that also lets users type their sentences and interact with some

[3] `https://en.wikipedia.org/w/index.php?title=Office_Assistant&oldid=901691073`, last access on Jun/2020.

[4] `https://en.wikipedia.org/w/index.php?title=Seaman_(video_game)&oldid=901233088`, last access on Jun/2020.

[5] `https://en.wikipedia.org/w/index.php?title=Google_Now&oldid=884721642`, last access on Jun/2020.

[6] `https://en.wikipedia.org/w/index.php?title=Google_Assistant&oldid=903618310`, last access on Jun/2020.

visual options, such as bubbles with suggestions of answers. *Google Assistant* can also learn information about its user and recover data from previous interactions in order to keep context (Purewal, 2016). Initially exclusive for Pixel and Pixel XL smartphones and available through messenger software *Google Allo* and *Google Home* speakers, it is now available for other *Android* and *iOS* devices.

Microsoft's personal assistant is called *Cortana*[7], after the video game character, released in 2014 for Windows Phone 8.1. Now available in various platforms, such as *Windows 10*, *Android*, *iOS*, and *XBOX OS*. This assistant is voice and text activated, and it can also learn from users' interactions and is able to keep context between queries (Kiseleva et al., 2016).

Also in 2016, *Amazon Alexa*[8] was released. Amazon's personal assistant was initially exclusive for *Amazon Echo* smart speakers, being able to set alarms, read audiobooks, play music, and connect to third-party services, such as *Uber*, through voice commands. Later, *Android* and *iOS* companion apps were released.

2.1.1 Chatbot Platforms

Some messenger platforms also opened their *APIs* for the development of chatbots. In these cases, the chatbot is accessed as a regular contact in users' contact lists, and users can chat with them just as chatting with another person, although in some cases there are a few other visual aids that will be further explained in Chapter 4 - "Communicative Strategies Investigation".

Several platforms are available for chatbot development, such as *Telegram*, which has a chatbot *API* since 2015. *WeChat*, a messenger app very popular in China also has chatbots, as well as *KiK* messenger (Candello and Pinhanez, 2016). Other platforms include *Skype* and *Slack*.

Besides platforms in which the chatbots will be available, like the ones mentioned above, there are yet other supporting platforms for chatbot development. Platforms such as *Blip* and *Chatfuel* can help chatbot developers to deal with aspects such as NLP and the dialogue itself. Works investigating such platforms are also available. For example, Galvão et al. (2019) evaluated *Chatfuel*'s usability by having users with diverse level of familiarity with technology, but no artificial intelligence background, use the platform to create chatbots.

[7] `https://en.wikipedia.org/w/index.php?title=Cortana_(software)&oldid=903554581`, last access on Jun/2020.

[8] `https://en.wikipedia.org/w/index.php?title=Amazon_Alexa&oldid=903387716`, last access on Jun/2020.

This work focuses on chatbots using Facebook Messenger as a platform for users to interact with. In 2016, Facebook released the Messenger platform API, allowing third parties to develop chatbots that could interact with Messenger users using the chat interface. In the case of Facebook Messenger, users can interact with chatbots both using natural language (by typing their sentences) and by clickable buttons and menus containing the chatbots' functions and suggestions of answers. There are many chatbots in many different fields, such as weather, shopping, news, among others (Candello and Pinhanez, 2016).

In the next section, we explore some works focusing on aspects of chatbots and conversational interfaces' design and evaluation.

2.2 Chatbot Design and Evaluation

Chatbots have been around for a long time. However, research supporting their design is more recent. Candello et al. (2017) realized a series of user tests exploring how different typefaces may influence users' perception of whether they are interacting with a chatbot or with a human through a messenger app. Portela and Granell-Canut (2017) made an experiment based on quantitative and qualitative methods in order to check how different chatbot behaviors (e.g. use of social cues, humor, and timing) may affect user engagement and affection towards the chatbot. They conducted user tests with questionnaires, semi-structured interviews, and both a chatbot and a Wizard-of-Oz experiment and saw that, as the current state of the art does not enable great conversations, most users were skeptical of having personal relations with chatbots, although some users considered that their experience using Siri was more empathetic because of the jokes and funny answers it provided. That way, they concluded that the combination of different behavior strategies may help to engage with users, but the level of human-likeness needs to be well balanced, so users will not get confused as to whether they are talking to a machine or a person.

Smestad and Volden (2019) investigated how the perceived personality of the chatbot impacts the user experience. The work argues that as people tend to anthropomorphize everything that acts or looks human, the same happens to chatbots, as users imbue human-like traits to them. Designers may use the chatbot's personality to influence the characteristics that will be associated with it in order to manage expectations users have on how it behaves. They made user tests using two chatbots each with a different level of personality (one agreeable and the other conscientious) and found out that personality has a significant impact on user experience, but that depends on context, function, and user group. In some cases, a more agreeable chatbot may have a better impact, while in other domains or user groups that personality may not be adequate.

Through a series of interviews with users of customer service chatbots, Følstad et al. (2018) concluded that users' trust in the chatbots is affected by elements related to the chatbot itself (such as how well it can interpret sentences; how human-like and polite it is; how it presents itself and what it can do; and how professional it appears to be, including spelling and grammar) and other factors related to the service context: as the company the chatbot represents, security and privacy, and the perceived risk associated to using the chatbot. So authors suggest that, in order to make their chatbots seem more trustworthy, designers should provide a reliable service to users; be honest regarding the chatbot's features and limitations; make the conversation more human-like, pleasant, and polite; make use of users' trust in the brand; and show that security and privacy are a priority to users.

In their survey, Piccolo et al. (2019) contrasted works on the chatbot domain to HCI predictions for the year 2020 from the year 2007. They point out that advances in Machine Learning and NLP, along with platforms that allow for easier development, brought the expansion of available conversational interfaces, in special textual chatbots (more than 300,000 of them on Facebook Messenger alone). Then they grouped works, such as case studies with first-time users, interaction style and evaluation, which tasks are appropriate for chatbots, trustfulness, conversation flow, and context. They finished indicating that there are many reports of users' frustration when using chatbots, so more user-centered research is necessary to address some limitations of the technology and help explore its potential and ensure its endurance.

Chaves and Gerosa (2019) also conducted a survey on chatbot works, analyzing 58 papers about text-based chatbots from several domains in order to check social characteristics that could be beneficial to chatbot interaction, as well as challenges and strategies for designing them. They grouped studies regarding desired social characteristics in a chatbot, such as conversational intelligence (characteristics that help managing interactions, such as conversational context and awareness of the discussed topic), social intelligence (habitual social protocols, such as responding to social cues, managing conflicts, or being emphatic), and personification (chatbot's perceived identity and personality, such as assigning personal traits to non-human agents, including appearance and emotional states). They also showed how these characteristics interact among themselves.

Jain et al. (2018) conducted a study with first-time chatbot users, asking them to use a set of chatbots for a period of time and then analyzing their conversation logs. The authors reached results similar to ours (particularly the need of clarifying the chatbot's capabilities, focus of this present work), showing which aspects the users did not like about the chatbots and which ones impressed them the most, indicating a possible way of improving chatbots design.

When looking at other types of conversational interfaces, like intelligent personal assistants, there are works with findings that can also be applied to text-based chatbots. Among these works, we can highlight a few, like the investigation of user expectations for personal assistants (Luger and Sellen, 2016), or the impact of interruptions on the user experience with a work-related assistant (Liao et al., 2016). On the same topic, Fischer et al. (2019) investigated how non-answer responses from a chatbot can break interaction, and how they can also lead to interaction recovery. Leahu et al. (2013) showed how users perceive Siri as human-like or robot-like depending on the subject. More recently, Clark et al. (2019) conducted a series of interviews to find out what users think is important in conversations with personal assistants. They discovered that people do not value the same characteristics when talking to conversational agents and when talking to other people, for example, users tend not to value developing a relationship with conversational agents, considering them just tools. So the authors suggest human-agent conversations should be seen as a new type of interaction, instead of just trying to mimic human-human conversations.

There are works describing methods and metrics for evaluating chatbots' components. For example, sentence accuracy and concept error rate for natural language understanding, and Common Answer Specification protocol to compare the chatbot's replies to a canonical answer for assessing dialog management (McTear et al., 2016b).

In their systematic literature review, Radziwill and Benton (2017) made a careful listing of quality issues and attributes for chatbots, as well as quality assessment approaches. They list 38 different quality attributes that, in general, are aligned with usability concepts of *efficiency* (such as graceful degradation, robustness to manipulation and unexpected input, etc.), *effectiveness* (interpretation of commands, general ease of use, among others), and *satisfaction* (giving conversational cues, entertaining the user, for example). It is interesting to note that they do not list any work considering communicability (either with this name or under a different one) as a quality attribute. In the same work, Radziwill and Benton (2017) also list quality assessment approaches found in literature, such as the PARADISE framework (Walker et al., 1997), among others, and propose a new approach based on the identified quality attributes.

However, to the best of our knowledge, there is little support for designers in deciding how their chatbots can (or should) present what they can communicate about and how to interact with them, i.e. what subset of natural language they are able to understand. In other words, there is little to support designers in improving their chatbots' communicability. In this work, we take a step in this direction, by identifying the communicative strategies that are currently being used by designers on their chatbots to convey their features to users.

* * *

Further related works, such as a description of the Semiotic Inspection Method and Pragmatics theories, are presented in chapters detailing these steps of the research for better reading. The Semiotic Inspection Method is presented in Chapter 4 - "Communicative Strategies Investigation", while the Pragmatics works are in Chapter 5 - "Pragmatics and Chatbots".

Chapter 3

Methodology

This chapter describes the methodology used throughout this work in order to identify the strategies being used to convey chatbot features to users. An overall methodology can be seen on Figure 3.1 and is comprised of four main parts: (1) bottom-up Semiotic Inspections, (2) top-down consolidation of strategies and sign classes, (3) Pragmatic analysis, and (4) user tests. In this chapter we give a general overview of the motivations, methods, and results for each of these parts. Further details for the methodology of each part are described in their respective chapters.

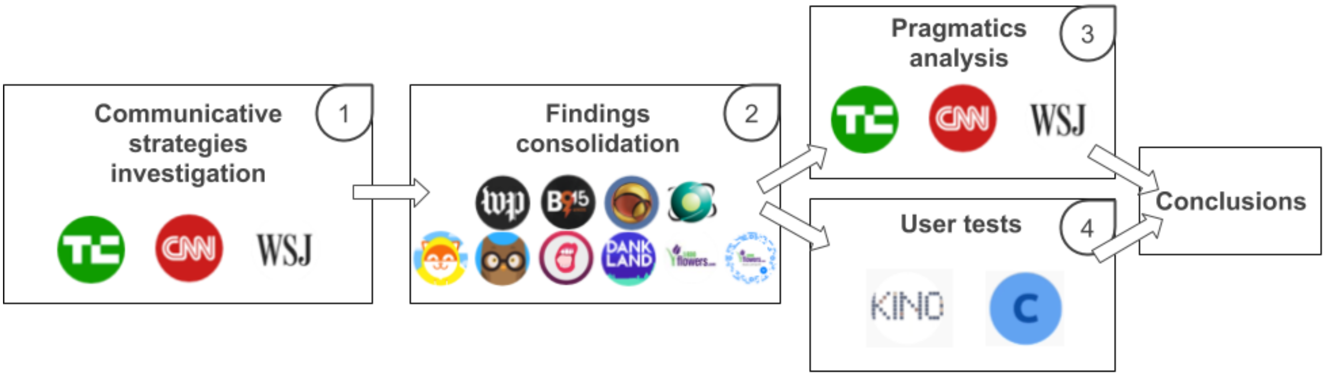

Figure 3.1: Overall methodology adopted in this work. Source: Authors.

3.1 Communicative Strategies Investigation

In order to explore how chatbots communicate their capabilities to users, we decided to explore a selection of chatbots using the Semiotic Inspection Method (SIM) (de Souza et al., 2006), which is based on the Semiotic Engineering (de Souza, 2005), an HCI theory that considers the software's interface as a form of designer-to-user communication. SIM can be used to analyze the underlying communication happening through the system's interface (de Souza, 2005; de Souza and Leitão, 2009). This designer-to-user communication includes *who* the designers believe the users are, what goals they want to achieve by using the system, and *how they are supposed to interact with the system in order to achieve these goals.* In order to apply SIM, the inspector has to perform a systematic inspection of the system.

These two characteristics of SIM (analysis of the metacommunication, including *how* users are supposed to interact with the system, and the systematic inspection), and the fact that it is not technology dependent and has already been used in a variety of different systems (de S. Reis and Prates, 2011) made us consider the method as appropriate to our objective of finding how chatbots' designers were conveying the chatbot's features to users.

We selected a small group of three chatbots with same domain (news) in order to have a homogeneous communicative context. With the intent to select chatbots that were considered of high quality, we selected chatbots that had been recently awarded. Also, that could be a sign that their designers made overall good choices during their development.

We then proceeded to carry out the scientific application of SIM in these three chatbots, in a bottom-up approach. Two researchers (including the author of the present work) with previous experience with SIM performed the inspections. Next, we triangulated the results between the researchers and, later, among chatbots, so we could ensure the scientific validity of the findings.

As result we were able to identify a series of six sign classes (visual elements aiding the conversation between chatbot and user) and eleven strategies for conveying chatbot's features to users. Other findings were related to the openness of the conversational interface's interaction space, namely, the different sentences a user can say to the system, and the approaches designers may take to tackle this issue. This first part of this work was first published in (Valério et al., 2017), its details are shown on Chapter 4 - "Communicative Strategies Investigation", and the results are on section 4.6 - "First Round Results: Identification of Sign Classes and Strategies in Chatbot Communication"

3.2 Findings Consolidation

The next step was to make a second round of chatbot inspections for consolidating the previously found sign classes and strategies. For that we took a top-down approach, in which the author (one of the inspectors that had applied SIM on the previous step) proceeded to make systematic inspections in order to find evidence of such sign classes and strategies in a set of ten other chatbots. This second set consisted of chatbots of diverse domains, including news, weather, humor, and feminism. Among these chatbots, three of them used Portuguese as language while the others were in English.

The results of that second round were the consolidation of the six sign classes and eleven strategies found in the first inspection, as we were able to identify them in the second set of chatbots. Another finding of that step were evidences of a 12th strategy in one of the second set's chatbot. Although we have evidences of that strategy, we still have not consolidated it yet. This part is also detailed on Chapter 4 - "Communicative Strategies Investigation", specially section 4.7 - "Second Round Results: Findings Consolidation", and it was first published in (Valério et al., 2018).

From that point, we had two questions to further investigate. We could (a) deepen the analysis of the chatbot communication, exploring Pragmatics' aspects of the chatbots, sign classes, and strategies. Or we could (b) test our previous findings in user-tests. As both were independent, we decided to pursue both questions. For (a) we needed to conduct an in depth analysis of chatbots in the light of some Pragmatics' aspects, while (b) would require creating two chatbots with different design approaches and conducting a series of user tests to check what users thought about them. We opted for investigating Pragmatics of chatbots first.

3.3 Pragmatics Analysis

We selected some theories that influenced the theory of Semiotic Engineering, and were already used in other Semiotic Engineering works to obtain in depth analysis of some aspects of the metacommunication, namely Searle's Speech Acts, Grice's Cooperative Principle, and Leech's Politeness Principle (de A. Barbosa et al., 2007; Lopes et al., 2019). The rationale was to make an in-depth analysis of the chatbot's messages using these theories. Another important aspect was to verify if these theories were compatible with a chatbot context, as they were originally intended for analyzing human-human communication, and we would use them for analyzing chatbot-human conversations. But as these works are used for analyzing metacommunications on Semiotic Engineering, it was a good sign of their applicability to chatbot context.

We decided to use the same set of chatbots from the first inspections, as they have coherent communicative context, and by having already applied SIM to them, we already had an initial analysis of their communication, as well as how it was presented (through the signs evidences). That would be ideal for an in depth analysis of these chatbots, and would allow us to see what that could add to the SIM analysis.

As results from this investigation, we have considerations about the use of these

Pragmatics' theories on chatbots, as they were originally intended for human to human communication, as well as evidences of the relation of some of them to the previously found strategies and sign classes. Besides that, our results indicate that these principles could be useful during chatbot design. We also found out that the Cooperative Principle can help explain communications breakdowns that happen during chatbot interaction. This part of the research is detailed on Chapter 5 - "Pragmatics and Chatbots".

3.4 User Tests

We also performed an evaluation comparing a chatbot based on the previously found strategies and sign classes to another one that did not take that into consideration. For that we used "Kino", a chatbot developed by D'Ávila (2018), who kindly let us use it to perform user-tests comparing different approaches to chatbot design. Kino was part of a work researching Natural Language Processing (NLP), so it relied on interpreting what the user typed instead of using menus and *quick replies*. That way, it had a very open conversational space. We proceeded to design a new chatbot called Cinemito, with the same features as Kino's, but having a different approach to interaction: focusing on closing its conversational space and including as many of the strategies and sign classes from previous findings as possible. Having both chatbots, we were be able to investigate the effects of two approaches to chatbot interaction: one open, using NLP, and other more restrictive, making no use of NLP, instead relying in some of the strategies identified in Chapter 4 - "Communicative Strategies Investigation".

Next we designed a test scenario that required the use of some of the chatbots' features, and elaborated a series of interview topics to be discussed with the participants in order to explore their opinions regarding some of the sign classes, strategies, and approaches used on both chatbots. As we were testing two systems, we also took precautions to avoid learning bias by inverting the chatbots order on every other test. Finally, we proceeded to recruiting users for the tests, and a homogeneous group of mostly STEM (Science, Technology, Engineering, and Mathematics) students accepted to participate in them.

During tests, the chatbots ran on a smartphone, using Facebook Messenger. The audio and smartphone screen were recorded, and, during tests, participants were asked to use the think-aloud protocol (Lewis, 1982) through the interaction with the chatbots. After using the chatbots, users were interviewed. The tests were conducted by the author who also also took notes of any unusual actions.

We analyzed the interviews, comments made by participants, and notes made during the evaluation. During the analysis, we grouped comments by common themes and checked for any interesting outliers.

The results from the analysis were good (and a few bad) reactions to some strategies and sign classes, as well as participants opinions about the two different approaches to the conversational design of the chatbots, specially regarding the openness of the conversational space and the relation between the use of NLP and users' perception of chatbot's intelligence. We were also able to identify advantages and limitations of the used sign classes and strategies used by the chatbots. Besides that, we also could relate some interaction problems that occurred to violations of the pragmatics principles. This part is detailed on Chapter 6 - "User tests".

Chapter 4

Communicative Strategies Investigation

This chapter explains the Semiotic Inspection Method in section 4.2 - "Semiotic Inspection Method" showing its necessary steps and situations in which it produces valid results. In section 4.3 - "Premises Regarding SIM Applicability on Chatbots" we explain the methodology for inspecting the chatbots, as well as the considerations we took when applying the Semiotic Inspection Method to chatbots, as it was the first time it was done, to the best of our knowledge. Then we explain the first (bottom-up approach, in section 4.4 - "First Round: Chatbots Selection and SIM Application") and second (top-down approach, in section 4.5 - "Second Round: Consolidation") rounds of inspections. We then show the results of the first round (section 4.6 - "First Round Results: Identification of Sign Classes and Strategies in Chatbot Communication") and of the second round (section 4.7 - "Second Round Results: Findings Consolidation"). Following that, in section 4.8 - "Discussion", we discuss our findings and possible treats to this work. Finally, section 4.9 - "Conclusions" contains our conclusions and insights from our experiences inspecting the chatbots.

4.1 Inspections Methodology

This part of the research was conducted in two stages. In the first one, we took a bottom-up approach, using SIM to inspect three similar-purposed chatbots. As a result, a set of six sign classes and 11 strategies used by chatbots' designers in their interface languages emerged. In order to consolidate these findings, in the second stage we selected a set of 10 other chatbots and, taking a top-down approach, we analyzed them using the sign classes and strategies as guides. The goal of these analyses was to register *if* and *how* the identified sign classes and strategies were used in the selected chatbots and whether there were any different sign classes or strategies that had not been identified yet. Figure 4.1 shows an overview of the adopted methodology.

We chose SIM for inspecting the chatbots because this method allows for a sys-

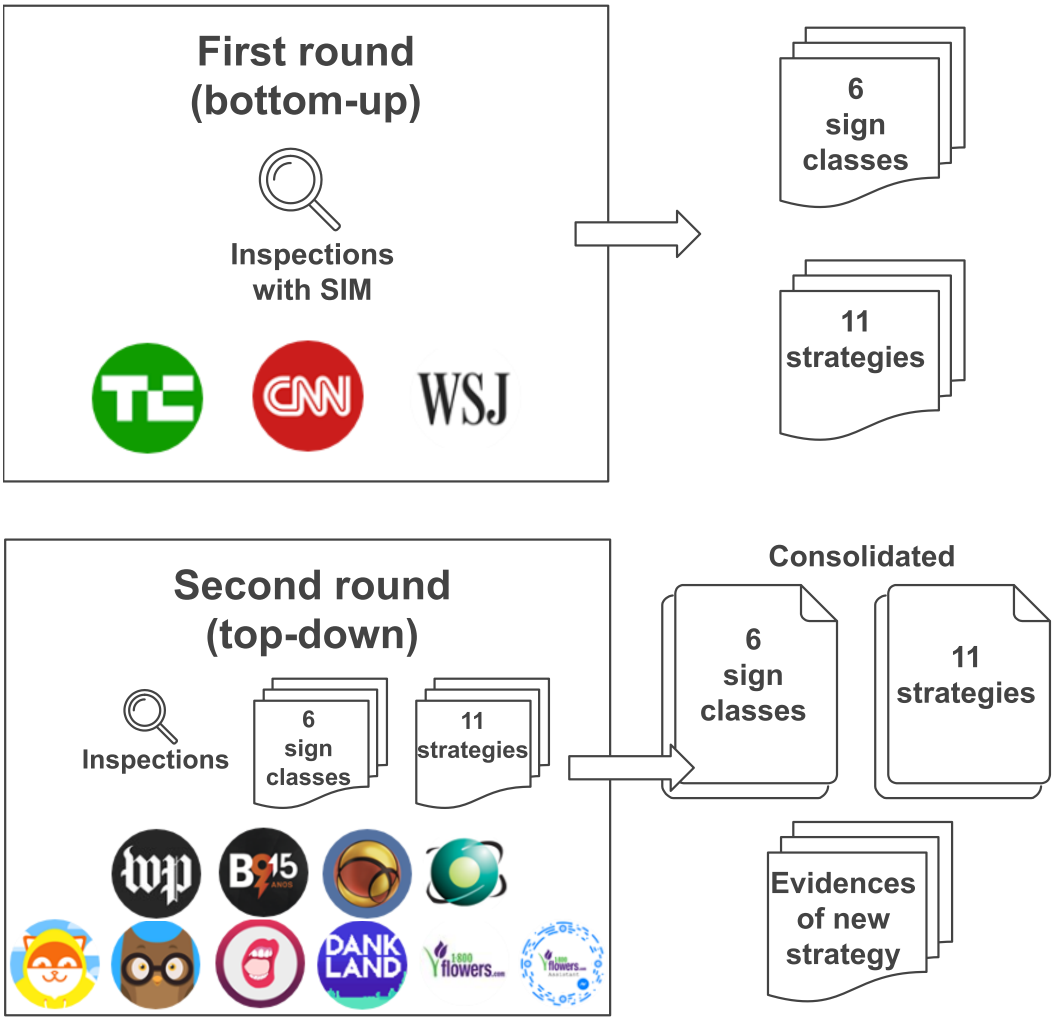

Figure 4.1: Inspections overview. Source: Authors

tematic analysis of the designers' metacommunication (de Souza et al., 2010), and, thus, does not depend on specific technologies or domains (de S. Reis and Prates, 2011). However, as it had not been applied before to conversational interfaces. This section explains the challenges faced during the SIM application and the premises adopted for using the method on chatbots. We then explain in more detail the methodology applied in each round of inspections performed in this work.

4.2 Semiotic Inspection Method

The Semiotic Inspection Method (SIM) is a qualitative evaluation method grounded on Semiotic Engineering – a Human-Computer Interaction (HCI) theory. Semiotic Engineering perceives an interactive system's interface as a designer-to-user communication. Designers communicate to users their design intentions and principles regarding the system they built through the system's interface. In other words, they communicate to users *who* the designers believe users are, *what* goals they expect users to want/have to achieve with the system, and *how* they are expected to interact with the system to do so. As users interact with the system itself, the designer's message is unfolded to them through the system. Thus, the system is considered to be a metacommunication artifact.

The property that defines how well the system communicates to users the designer's intentions and underlying principles have been defined as communicability (de Souza, 2005; de Souza and Leitão, 2009). Thus, in order to assess this property, communicability evaluation methods have been proposed. One of these methods is the Semiotic Inspection Method (de Souza et al., 2006, 2010). SIM is an inspection method based on system interface analysis conducted by specialists. It allows for a systematic inspection of the system's interface, reconstructing the designer's intended metacommunication and identifying potential inconsistencies and communication problems (de Souza et al., 2006).

There are two phases to SIM: preparation and execution. During preparation, the specialist defines the inspection goals, chooses the system to be inspected and performs an informal evaluation, defines the focus and scope of the evaluation, and describes a scenario that will guide the inspection.

In the execution phase, the specialist follows the method's five steps:

(i) In the first step, the specialist analyzes the system's metalinguistic signs and reconstructs the system's metacommunication message based only on this type of sign. Metalinguistic signs "explicitly communicate to users the meanings encoded in the system and how they can be used" (de Souza and Leitão, 2009, p. 19), i.e. they are signs that explain other signs, such as documentation text, error messages, tooltips, and others.

(ii) In the second step, the specialist does the same for static signs. Static signs are signs that can be "interpreted independently of temporal and causal relations" (de Souza and Leitão, 2009, p. 19), i.e. they can be seen on the interface at a single moment in time, such as layout, toolbar buttons, among others.

(iii) In the third step, the specialist reconstructs the system's metacommunication message once again but based on dynamic signs only. Dynamic signs "are bound to temporal and causal aspects of the interface, namely, to interaction itself" (de Souza and Leitão, 2009, p. 19), i.e. they represent sequences of actions and system behavior, and confirm (or not) the users' anticipation about the interaction.

These steps are done in order but can be revisited iteratively by the specialist (de Souza et al., 2010).

The first three steps generate a segmented analysis of the interface, whereas the following two steps integrate them.

(iv) In the fourth step, the specialist contrasts the three metacommunication messages reconstructed during the segmented analysis and compares them looking for inconsistencies and potential problems.

(v) Finally, in step five, the specialist consolidates the metacommunication messages generated in each step and assesses the system's communicability as a whole.

SIM can be used in scientific contexts to generate valid knowledge in HCI (de Souza et al., 2010) (which is the case of the present work, as detailed in sections 4.4 - "First Round: Chatbots Selection and SIM Application" and 4.5 - "Second Round: Consolidation"). To do so, two other steps must be considered when applying the method. During the preparation phase, it is necessary to define the research question researchers are interested in. Also, after the application, a triangulation step is added to the analysis. Triangulation involves the generation of other results (e.g. by other specialists or through compatible methods) that can be used to consolidate or discuss the results obtained through SIM.

4.3 Premises Regarding SIM Applicability on Chatbots

SIM focuses on the communicative aspects of the intended metamessage sent from designers to users. By supporting the reconstruction of the metamessage from the system's designers, SIM allows for the identification of communicative strategies used by designers to communicate their design intents and principles (de Souza and Leitão, 2009). It has been shown that it can be applied to a variety of contexts and technologies (de S. Reis and Prates, 2011). For instance, SIM has been used on educational software (de S. Reis and Prates, 2011), human-robot interface (Bento et al., 2009), the audio aspect of video games (Corrêa et al., 2012), and online social networks (Valério et al., 2016). As SIM is not dependent on conventional visual interfaces, we decided to use it to evaluate the communicability of chatbots. However, to the best of our knowledge, it was the first time it was applied in this context. Therefore, we had to adopt some premises for the inspections.

Differently from other domains and technologies, SIM has been applied to, chatbot interaction is based mainly on natural language message exchange, simulating a human-human conversation. Therefore, our first challenge in order to apply SIM to this context was to check whether Semiotic Engineering's definitions of what should be considered metalinguistic, static, and dynamic signs held in the context of conversational interfaces.

As explained in the previous section, metalinguistic signs are those that explain to users the meaning of other signs with which they interact; static signs are those that can be interpreted independently from temporal and causal relations (de Souza et al., 2006; de Souza and Leitão, 2009; de Souza et al., 2010). In traditional graphical interfaces, static and metalinguistic signs typically make use of different signification systems. Static elements are usually represented by interface widgets, such as buttons, menus, and displayed options, while metalinguistic signs usually make use of natural language to explain other signs, and are usually presented through tooltips, warning, or error messages, and the help section.

In a chatbot, however, a natural language utterance can either be about a topic of the chatbot's conversation, or it can be used to explain other signs or even the chatbot itself. Thus, in the former situation, the utterance would be a static sign, whereas in the latter it would be considered a metalinguistic sign. Hence, in this context, these two types of signs are syntactically similar, for they are conveyed through the same signification system – natural language. Therefore, in order to differentiate these two types of signs, it is necessary to carefully analyze the meaning and context of the message.

In our analysis, every chatbot's utterance describing the chatbot itself or its capacities was considered a metalinguistic sign. Fig. 4.2 on the next page illustrates an example of a metalinguistic sign on CNN's chatbot in which it informs the user about how to access some of its features. Also, as in the application of SIM in other contexts, we also considered news stories about the chatbot (e.g. a story about its release), and any other text describing the chatbot or its features on its website as metalinguistic signs.

Utterances about conversational topics that the chatbot could cover were considered static signs (that is, about topics other than the chatbot itself or its features). Fig. 4.3 on the following page is an example of static sign on Poncho chatbot, in which it responds to a user utterance ("tell a joke") with a joke, that clearly is a conversation on a topic not about the chatbot itself, and is, therefore, considered a static sign.

Furthermore, visual elements used by the chatbot in combination with text to support the conversation with the user (e.g. *persistent menus*, *cards*, or *quick replies*[1]) were also considered static signs. Fig. 4.2 on the next page also contains examples of static signs, such as the *quick replies* (the blue "bubbles" on the bottom of the image) hinting at possible follow-ups the user may select, in this case, they are "Editor‘s Picks", "Topics", and "Unsubscribe".

[1]The classes of visual elements identified in our analysis are described in detail in subsection 4.6.1 – "Sign Classes".

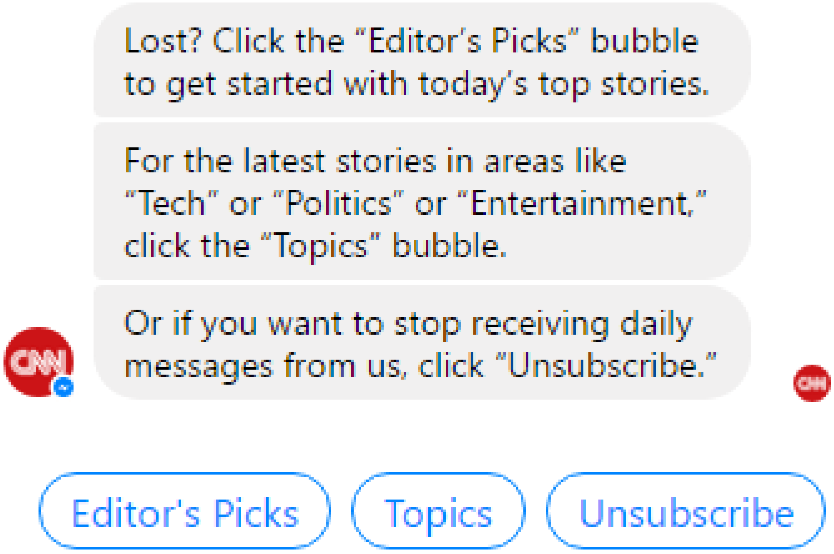

Figure 4.2: Example of metalinguistic sign on the CNN's chatbot. Source: Authors

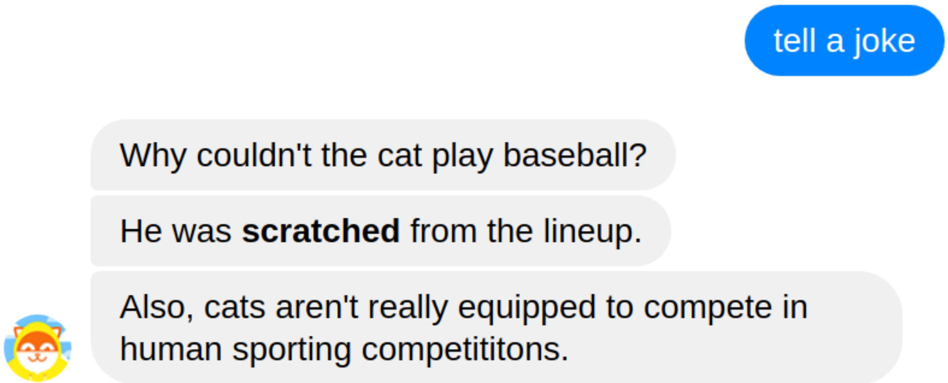

Figure 4.3: Example of static sign on the Poncho chatbot. Source: Authors

Dynamic signs are defined as signs that are bound to temporal and causal relations, and represent the interaction itself (de Souza et al., 2006; de Souza and Leitão, 2009; de Souza et al., 2010). In chatbots, dynamic signs are represented by the conversation – i.e. the exchange of messages itself, that is the chatbot's behavior which consists of the transitions between states of the chatbot and is bound to temporal or causal relations.

4.4 First Round: Chatbots Selection and SIM Application

For the first part of this work, we decided to select a small set of chatbots from a single domain and that were considered to be successful. The small number of chatbots was to enable a detailed in-depth qualitative analysis of each one of them. The single domain aimed at focusing on a more homogeneous communicative context. Finally, their success was taken as an indicator that their designers made overall good decisions during the design process.

Hence, we decided to initially inspect three news-related chatbots on Facebook Messenger platform, the winners of 1st, 2nd and 3rd places of the news category on Chat-Bottle Awards 2017[2]. The selected chatbots were: TechCrunch[3], CNN[4], and the Wall Street Journal[5] (WSJ) chatbots. Although all three chatbots are news-related, they focus on different content. While CNN chatbot covers a wider range of topics, TechCrunch focuses on tech and start-up stories, and WSJ is business-oriented.

We inspected the chatbots using the scientific application of SIM (de Souza et al., 2010). Our research question was: "What communicative strategies have been used by popular chatbots to convey their features to users?". Similar evaluation scenarios were created for guiding the inspections of each chatbot. The main difference between the scenarios was the topic of the conversation since each chatbot focused on different news content. The considered scope was the whole chatbot, for the analyzed chatbots did not have many different functions.

The inspections were performed by the author and one other MSc student. Both of them had previous experience with the method, both in academic and research contexts. They were responsible for applying SIM in each chatbot separately. After that, the results of the inspections of the two researchers for each chatbot were triangulated. Finally, the results for each chatbot were triangulated with the other's chatbots results. That was done to ensure the scientific validity of the results. The researchers did not talk to each other about their own inspections until the triangulation. That was necessary to avoid one researcher influencing the other. The results were then discussed with other specialists. These inspections were conducted during June 2017 and were published in (Valério et al., 2017).

As a result of these inspections, we were able to identify six sign classes used as

[2]`https://chatbottle.co/awards/1st-chatbottle-awards-2017-winners`, last access on Jun/2020.

[3]`https://www.messenger.com/t/techcrunch`, last access on Jun/2020.

[4]`https://www.messenger.com/t/cnn`, last access on Jun/2020.

[5]`https://www.messenger.com/t/wsj`, last access on Jun/2020.

visual cues on the chatbots' interface and 11 strategies related to presenting the chatbots' features to users. These results are shown in section 4.6 - "First Round Results: Identification of Sign Classes and Strategies in Chatbot Communication"; the visual classes are shown in 4.6.1 - "Sign Classes" subsection, while the strategies are explained in 4.6.3 - "Strategies For Conveying Features" subsection.

4.5 Second Round: Consolidation

In the second stage of our analysis, our goal was to consolidate the identified sign classes and strategies by investigating if other chatbots made use of them as well and whether any other sign classes or strategies would emerge. To do so, we broadened the set of analyzed chatbots by number and domain and took a top-down approach. Thus, we selected 10 new chatbots, four from the same domain as before (news) and the others distributed in different domains, namely: weather, entertainment, sales, environmentalism, and feminism.

In each chatbot we performed a systematic inspection, registering which sign classes and strategies (previously identified) they used and how, and if any other sign class or strategy that had not been identified in the first stage of our research would emerge. Note that, although a systematic analysis inspection was performed, we did not conduct a complete application of SIM – the metalinguistic, static, and dynamic signs were analyzed, but the metamessage was not reconstructed or analyzed as a whole.

We selected Brazilian and International (English-speaking) chatbots that were either well-known or popular[6]. Next, we list each one of the 10 selected chatbots, presenting their names, abbreviation, URL, domain and a quick description, and the language they use.

- **Washington Post (WP)**[7]: a news chatbot that focuses on political stories from the USA; in English;
- **Brainstorm 9 (B9)**[8]: a Brazilian news chatbot that focuses on news about communication, culture, and media; in Portuguese;

[6]The chatbots were identified in several different sources, varying from chatbots used as examples in scientific papers, to those cited or awarded in chatbot sites, such as Chatbottle (`https://chatbottle.co`) or local Brazilian groups dedicated to chatbots.

[7]`https://www.messenger.com/t/washingtonpost`, last access on Jun/2020.

[8]`https://www.messenger.com/t/brainstorm9`, last access on Jun/2020.

- **UOL Notícias (UOL)**[9]: another Brazilian news chatbot with a wide range of news topics; in Portuguese;
- **BOL (BOL)**[10]: another new chatbot from Brazil, which serves as a FAQ (Frequently Asked Questions) for a Brazilian ISP (Internet Service Provider) and also sends news stories to users; in Portuguese;
- **Poncho (PNC)**[11]: a very irreverent chatbot that informs users about the weather and tells a lot of jokes; in English;
- **Smokey (SMO)**[12]: a chatbot that aims at creating awareness about air pollution and also lets users know about air quality in various cities around the globe; in English;
- **Beta (BET)**[13]: a Brazilian chatbot that focuses on keeping users up to date about feminist matters in Brazil; in Portuguese;
- **Dankland (DNK)**[14]: a chatbot that can create *memes* out of figures the user sends to it; in English;
- **1-800-Flowers.com (18F)**[15]: chatbot related to the *1-800-Flowers.com* online store, focusing on customer service; in English; and
- **1-800-Flowers.com Assistant (18FA)**[16]: another chatbot for the online flower store, but focused on selling flowers and bouquets through chat; in English.

The second round of inspections took place in January 2018. It was completed by the author. The inspections' results were then discussed with the other specialists. The results of this stage of the research are presented in section 4.7 - "Second Round Results: Findings Consolidation" and were published in (Valério et al., 2018).

[9]`https://www.messenger.com/t/UOLNoticias`, last access on Jun/2020.

[10]`https://www.messenger.com/t/BOL`, last access on Jun/2020.

[11]`https://www.messenger.com/t/hiponcho`, last access on Jan/2018.
Unfortunately, Poncho was discontinued in May/2018 (`https://www.theverge.com/2018/5/29/17404650/poncho-weather-shutting-down-betaworks-dirty-lemon`).

[12]`https://www.messenger.com/t/smokeybot`, last access on Jun/2020.

[13]`https://www.messenger.com/t/beta.feminista`, last access on Jun/2020.

[14]`https://www.messenger.com/t/dankland`, last access on Jun/2020.

[15]`https://www.messenger.com/t/1800flowers`, last access on Jun/2020.

[16]`https://www.messenger.com/t/1800FlowersAssistant`, last access on Jun/2019.

4.6 First Round Results: Identification of Sign Classes and Strategies in Chatbot Communication

This section presents and discusses the results of the first round of analyses in which three chatbots were inspected to identify how designers were communicating the chatbots' features to users. As a result, we have identified visual sign classes that are being used in this communication and potential breakdowns that could be associated with their use, as well as the communicative strategies adopted by their designers.

4.6.1 Sign Classes

As mentioned before, chatbots differ from traditional graphical user interfaces especially because of their text-based natural language input and output. This means designers (might) have fewer options of visual cues to choose from when designing chatbots as compared to traditional user interfaces. However, chatbot platforms – such as Facebook Messenger, Telegram, and others – offer chatbot's designers a set of possible inputs and outputs apart from textual messages. Chatbots may send figures, offer suggestions of replies, or even show a menu to users.

Even though the three chatbots from the first round of inspections have a similar purpose (sending daily news to their users), they offered users different interactive possibilities. Through our analysis, we identified the **types** of signs that were used in the chatbots' interface language presented to users. They represent visual cues used by chatbot designers to convey or reinforce different ways to interact with the system. In this section, we present the six classes of signs identified, and for each one of them, we explain the class and show an example.

- **C1** – ***Simple Message***: a message with text and/or emoji. On Facebook Messenger, this type of sign is represented by a gray rounded rectangle around a black text if the message was sent by the chatbot, or by a blue rounded rectangle around white text if the message was sent by the user. Fig. 4.4 shows a *simple message* sent by the TechCrunch chatbot as a response to another *simple message* sent by the user.
- **C2** – ***Simple Image***: a message with an image. The image may be static or animated. On Facebook Messenger, this type of sign is represented by the image

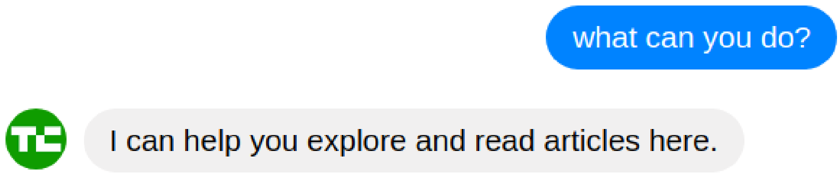

Figure 4.4: TechCrunch's *simple message* example. Source: Authors

itself with a rounded border and a subtle gray outline. Besides that, by the right side of the image, there is a button for forwarding the image – a blue icon on the app or a gray arrow pointing up on the web version. If the image was sent by the chatbot, there will be a small gray emoji on the bottom right-hand side of the image for reacting to the message. Otherwise, if the user sent the image to the chatbot, the gray emoji will not be shown. Fig. 4.5 shows the TechCrunch chatbot replying a *simple image* followed by a *simple message* on Messenger App.

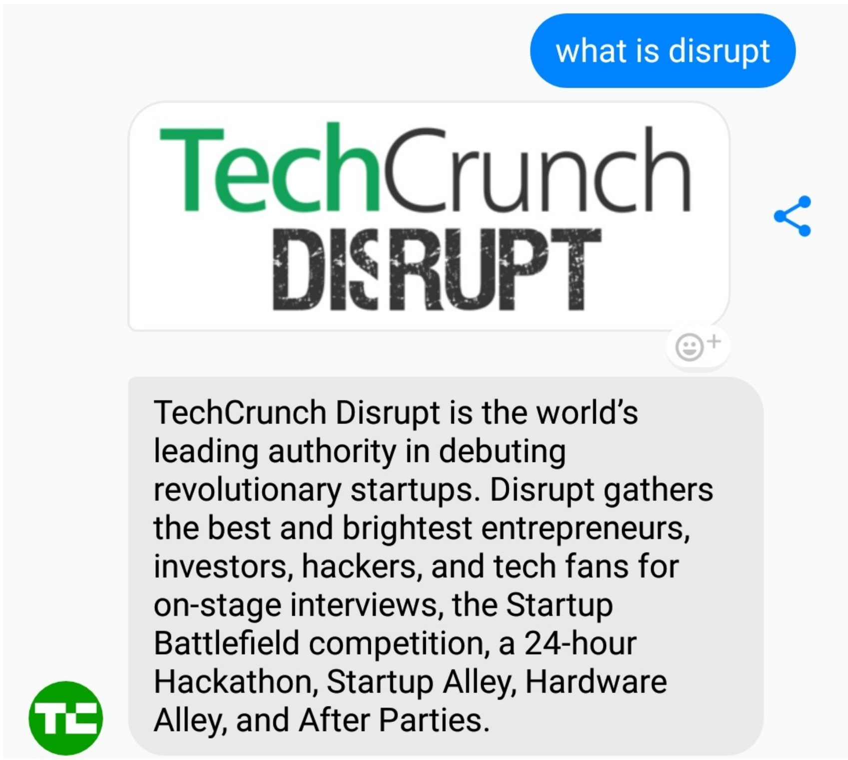

Figure 4.5: TechCrunch's message containing a *simple image*. Source: Authors

- **C3** – ***Suggestions* or *quick replies***: these are buttons that suggest messages the user can send to the chatbot. When the user clicks on a suggested message, that message will appear on the chat history as if it was typed by the user themselves, but the other suggested messages (if any) will vanish from the interface. In other words, *quick replies* are not permanent, as they do not remain in the chat history and disappear once the chatbot state is changed, either by the chatbot itself (by sending another message, for example), or by the user interacting with the chatbot by selecting a *suggestion*, typing a message, or taking another action (see Figure 4.7, the top image shows all of the *quick replies* available to the user, the bottom image shows the history after the user selected the "1 PM" option). Facebook Messenger represents *suggestions* with a blue outline. Fig. 4.6 shows *suggestions* of some stories the CNN chatbot can show the user: "Editor's Picks", "News", "Politics", and "Business".

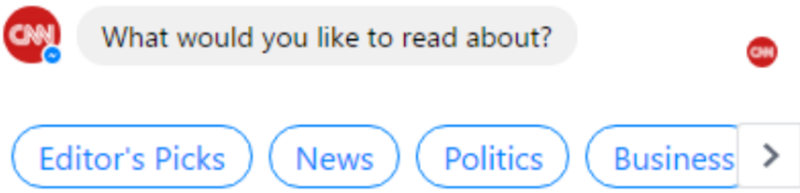

Figure 4.6: CNN's *suggestions* example. Source: Authors

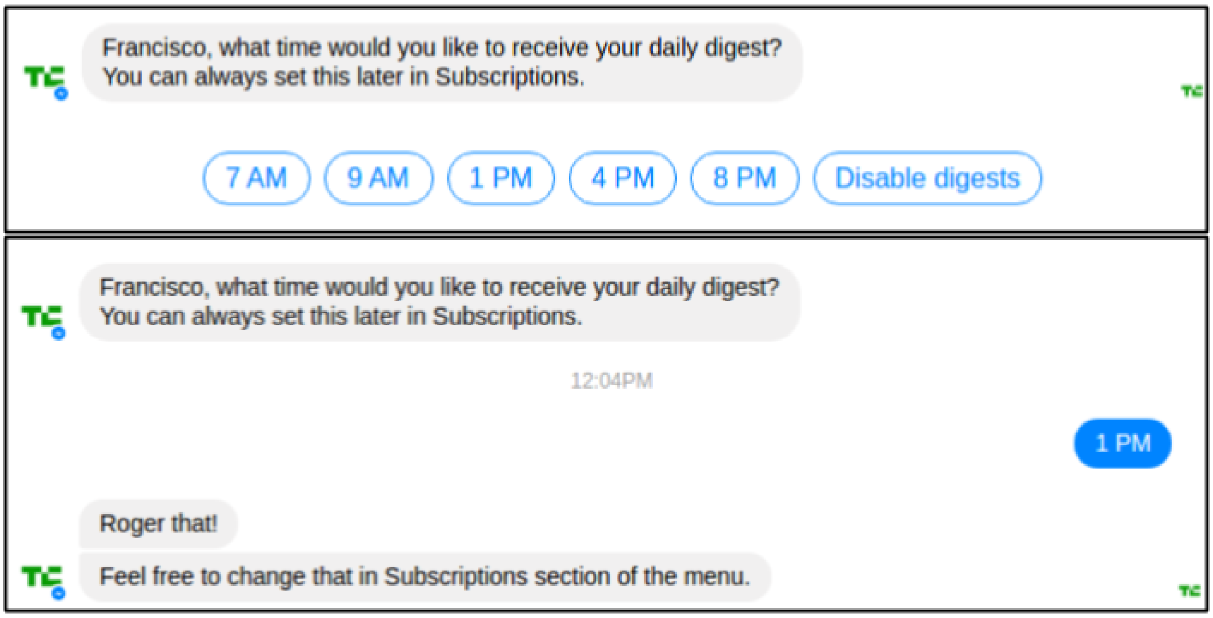

Figure 4.7: TC's *suggestions* before being selected (top) and after (bottom). Source: Authors

- **C4 – *Card***: a set of pieces of information to the user and/or actions users can take. These *cards* are permanent (as opposed to *quick replies*) since they remain in the chat history and do not disappear upon further interactions. This allows users to go back and explore other options they initially did not choose. Facebook Messenger represents *cards* with a gray outline, and each action is represented by

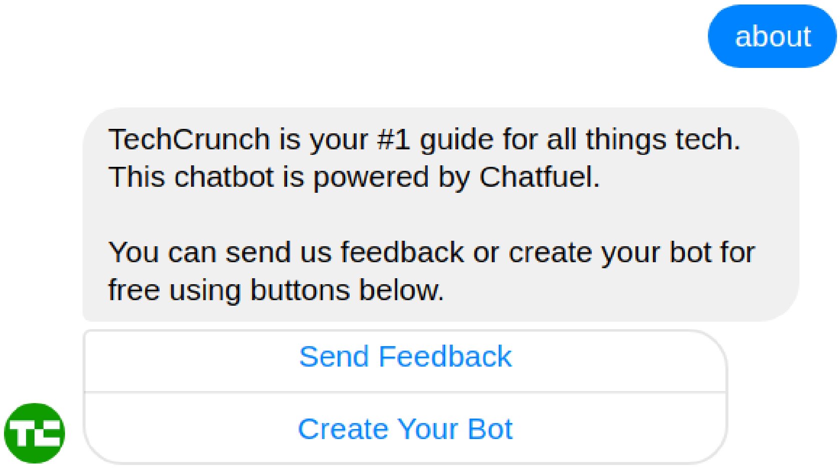

Figure 4.8: TechCrunch's *card* with no associated topic nor image. Source: Authors

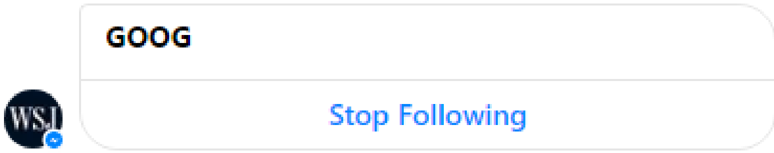

Figure 4.9: WSJ's *card* with a topic (GOOG) and no image. Source: Authors

Figure 4.10: TechCrunch's *card* with an image. Source: Authors

a button with a blue text. Fig. 4.8 shows the TechCrunch chatbot offering the user a *card* with two options: "Send Feedback" and "Create Your Bot". These buttons can stand by themselves, be grouped by topic, or even be associated with an image (with or without an external link). Fig. 4.9 shows an example of a *card*

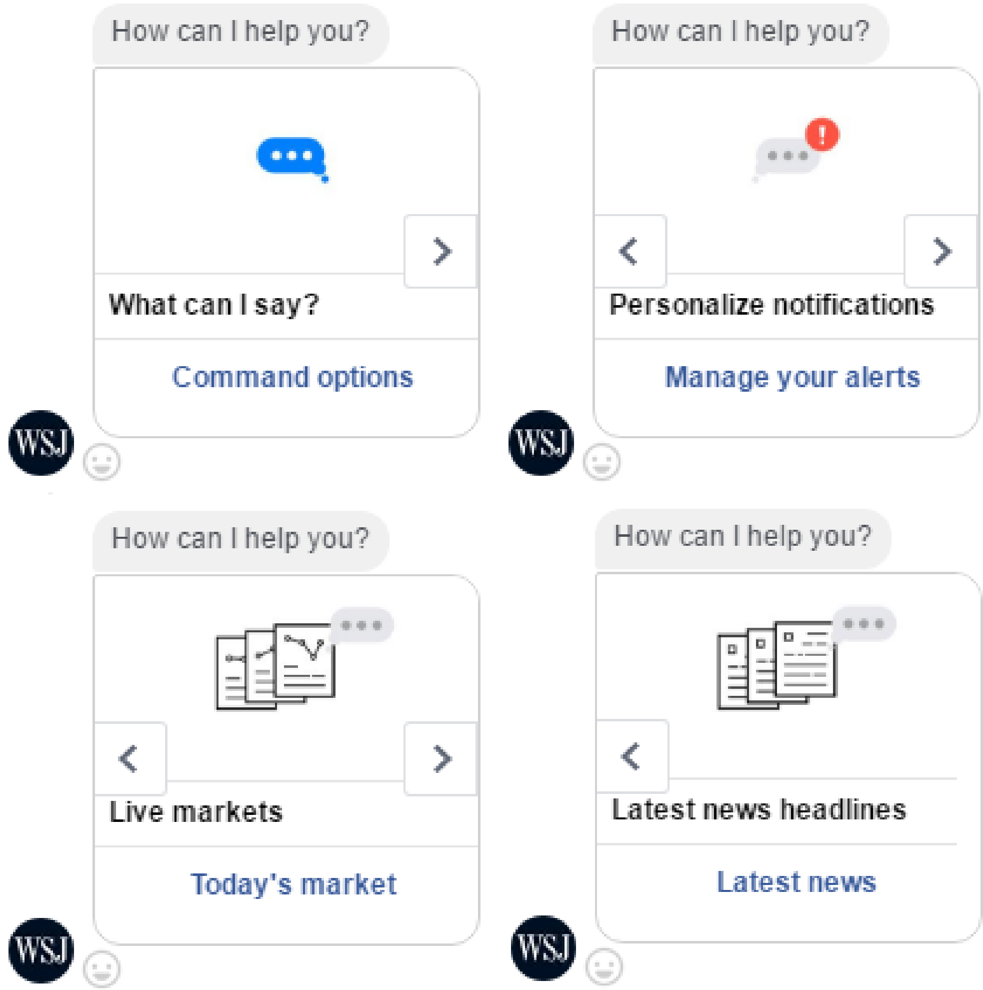

Figure 4.11: WSJ's main menu *carousel*. Source: Authors

with the topic "GOOG" (a ticker symbol on the stock market) and the option "Stop Following", whereas Fig. 4.10 shows an image of a *card* associated with a piece of news from TechCrunch.com – the title is in black, and there is only one action (button) associated with it: "View on Web" (in blue). The user can actually view the news by either clicking the image or the button.

- **C5 –** ***Carousel***: a collection of *cards* that allows the user to "flip through" different *cards*. Fig. 4.11 shows the *carousel* WSJ uses to show its main menu: a set of *cards*, each with its own image and buttons depicting the many features the chatbot can offer. The user can flip through different *cards* by clicking on the arrow buttons that show up on the sides of the *card* when hovering, on Messenger Web; and by sliding horizontally, on the smartphone App.

- **C6 –** ***Persistent Menu***: a set of buttons the user can access at any time. Fig. 4.12 shows the *persistent menu* for CNN on the Messenger App interface, while Fig. 4.13

shows the *persistent menu* for CNN in the Messenger Web interface. Both versions of the menu show the “Editor’s Picks”, “Topics”, and “Help” entries, while only the app version has the “Send a Message” option. On the web interface, users may type their messages directly on the text-input box under the menu. On the app, however, they must click ‘Send a message’ first and then type.

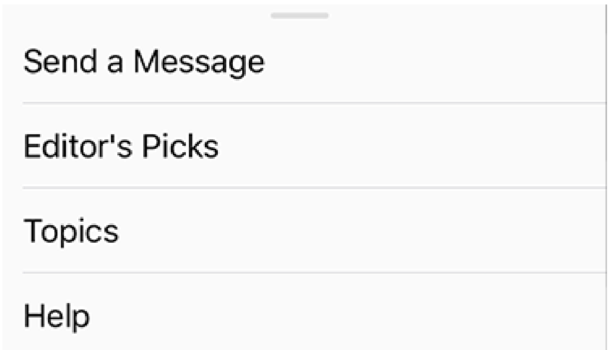

Figure 4.12: CNN’s *persistent menu* on the mobile app. Source: Authors

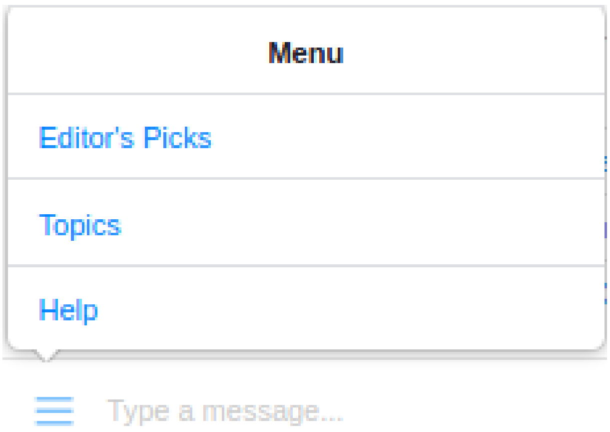

Figure 4.13: CNN’s *persistent menu* on Messenger’s web interface. Source: Authors

4.6.2 Sign Classes Considerations

The first two sign classes presented (*Simple message* and *Simple image*) represent types of messages that both users and chatbots can generate and exchange with each other. The other four sign classes (*Suggestion*, *Card*, *Carousel*, and *Persistent menu*) provide users with an indication of possible productive communicative paths they can take. By offering users options on topics or utterances, designers present to users some of the meaningful directions the conversation can take – i.e. topics or utterances the chatbot is prepared to understand and answer about. The options also make it easier for users to interact (since it spares them from typing). On the other hand, those four sign classes minimize designers' work on creating multiple scenarios and conversation flows. By employing these classes, the chatbots work closer to the usual graphical interface and less as a natural language interaction experience.

Once the user selects a predetermined action (a *suggestion* or a button in a *card* or the *persistent menu*), Facebook Messenger represents it as if the user had typed it themselves: with white text in a blue rounded rectangle. There is no visual difference in the dialog history between the resulting action of selecting a *suggestion* or typing the message. Initially, one could think that the feedback of selecting a *suggestion* could be interpreted as "by selecting this action I am saying *[action]* to the chatbot". This could help the user understand the outcome of the action, i.e. "I have got this result because I said *[action]*". However, this is actually deceiving, since the same message can be interpreted by the chatbot in different ways depending on how the user entered the input (by typing or selecting a *suggestion*). Thus, representing the message the same way but allowing for different interpretations from the chatbot can cause communication breakdowns.

Fig. 4.14 shows an example of this: on the left side we can see a dialog in which the user selects an option on the card ("Manage Subscriptions"); on the right side, we can see a dialog that starts the same way, but the user types the text instead (exactly the same text as the button on the *card*: "Manage Subscriptions"). However, each dialog has a different outcome: apparently, the chatbot lost the context when the user typed the message, even though its content is the same as what was suggested in the first place. As both user utterances are represented in the same way, the user may be led to believe they are the same. Nonetheless, the chatbot understands them differently. Users may never notice the potential ambiguity of their utterances, limiting their understanding of their capability to express themselves.

Another point we would like to emphasize is that *suggestions* and *cards* work in different ways: as mentioned before, *suggestions* disappear from the chat history as soon as the user selects one; on the other hand, the options presented on a *card* will stay on that

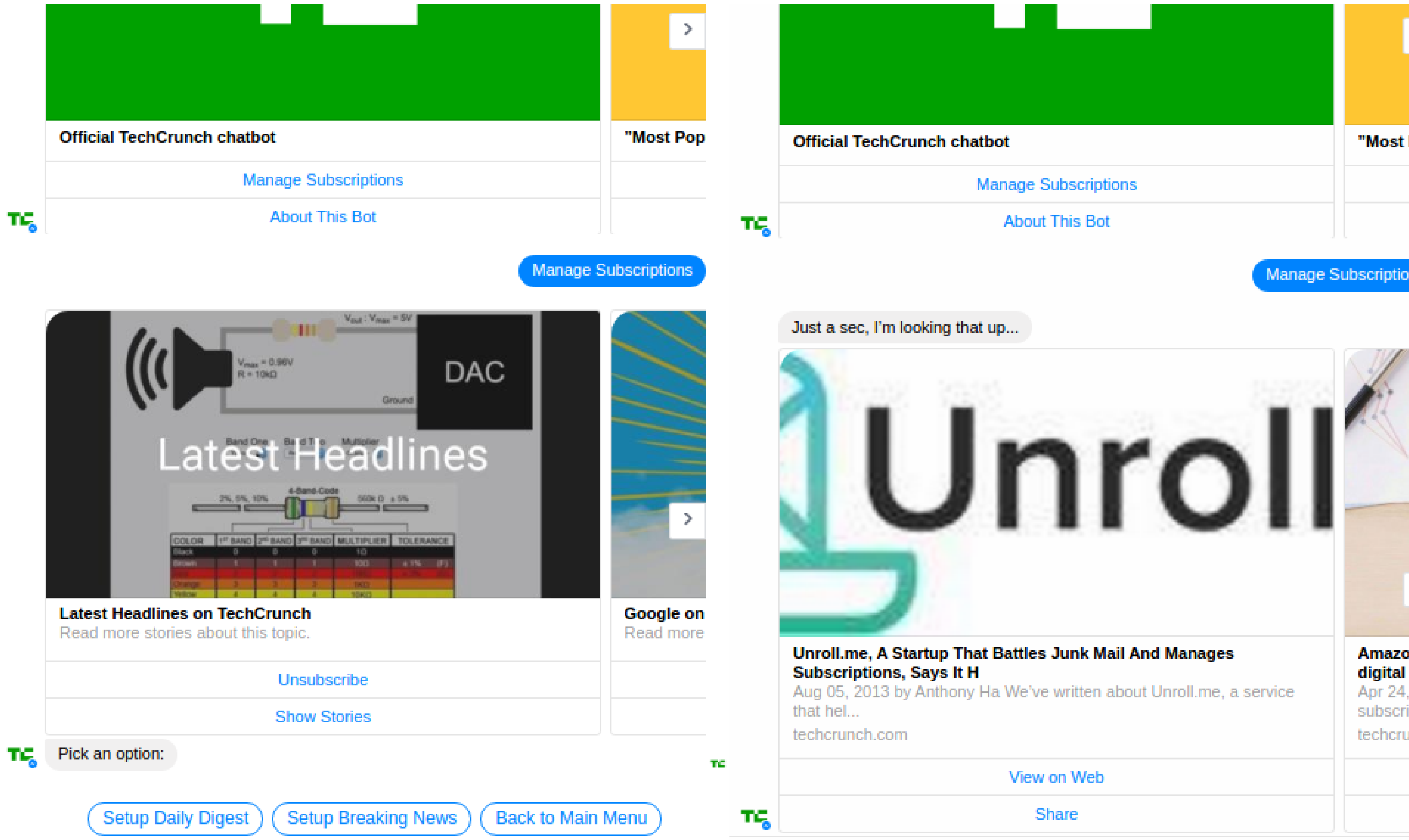

Figure 4.14: Left image shows TechCrunch's feedback when users choose the *suggestion* 'Manage Subscriptions' presented; Right image shows TechCrunch's feedback when users type 'Manage Subscriptions'. Source: Authors

card after an option is selected and, thus, in the chat history. Thus, it would make sense to use *suggestions* when the designer believes the options being offered to be mutually exclusive interactive paths at that point. *Cards*, in their turn, would be preferable if designers want to allow users to have the possibility to explore the various options, as users could go back on the chat history and select one of the other options.

As mentioned before, *cards* may or may not have images. When they do, sometimes the images have external links and sometimes they do not (CNN does not use links on images, for instance). When an image in a *card* is associated with an external link, it shows a URL in gray below the black title (as in Figures 4.10 on page 40 and 4.14). These figures also show how TechCrunch uses buttons on *cards* as redundancies to links on images. WSJ, however, does not adopt this strategy: when it uses images with links on *cards*, it does not associate redundant buttons with them. Thus, users may not notice or understand that there is a link associated with the image in that *card*, potentially causing communication breakdowns.

The *persistent menu* is the main difference between Web and App versions of

Facebook Messenger. On the Web version, the menu can be opened by clicking on the hamburger icon beside the textbox where users type their messages (Fig. 4.13 on page 42 shows the icon on the bottom left). On the App version, the menu can be made always accessible by default (Fig. 4.12 on page 42), and users have to click on "Send a message" to be able to type something. This inversion (between clicking and typing orders) is crucial for the user experience with the chatbot. It is much more inconvenient to type messages on the App since you have to select "Send a message" before typing. On the Web version, on the other hand, the menu can go unnoticed, as it is more discrete.

During more recent inspections, we noticed that the way the "Send a message" option on the *persistent menu* on the Facebook Messenger app is presented has changed. While on the previous version (in June 2017) "Send a message" is depicted as just one of the set of options on the *persistent menu* (Fig. 4.12), almost as if it was a decision made by the chatbot's designer for the menu; the new version changed the text for a "Send a message..." with suspension points ("...") and a dark gray font in a light gray rounded rectangle (Fig. 4.15), distinguishing it from the other options on the *persistent menu*.

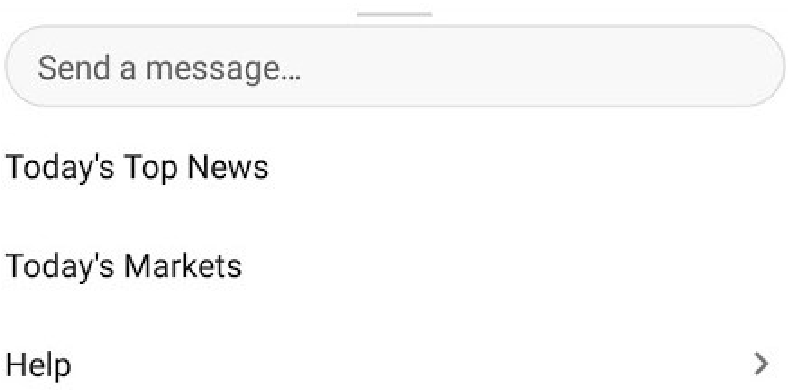

Figure 4.15: WSJ's persistent menu on messenger app on January 2018. Source: Authors

Finally, it is important to note the different terms we use when referring to menus. *Persistent menus* are the ones that are always available to the user (as defined above in this subsection). Main menus are accessible through chat messages and are displayed on the chat history. They may (or may not) be redundant. We can see the difference between both menus on WSJ: Fig. 4.15 shows WSJ's *persistent menu*, in which we can see that the main actions users can take are "Today's Top News", "Today's Markets", and "Help"; in Fig. 4.11 on page 41, the options in the main menu are "What can I say?", "Personalize notifications", "Live markets", and "Latest news headlines". This means that, for WSJ, the main menu is more focused on helping users understand the purpose of the chatbot and how it works; and the *persistent menu* acts as a shortcut to often-used features. Not all other chatbots adopt this approach.

4.6.3 Strategies For Conveying Features

In this section, we present and discuss the strategies used by chatbots to inform their features to users that resulted from our analysis. The strategies are the following:

- **S1 – Showing the main feature on the first message.** One of the strategies we identified was to use the first message to greet the user and present the feature the chatbot considers as the most important one. By doing this, it prevents the user from being lost and having to ask the chatbot what it can do.

  Two of the analyzed chatbots use this strategy: TechCrunch and WSJ. Both chatbots let the user know right off the start that their primary goal is to send daily messages, even though they do have more features. The choice of making the daily messages their most important feature is interesting since it ensures that users would continue to hear from the chatbot, creating a longer-lasting bond. Therefore, even if the user forgets about the chatbot after some interaction, the daily message will remind them that the chatbot is still there.

- **S2 – Guiding the user through a short tutorial during first messages.** In this strategy, the first conversation between the chatbot and the user includes a short tutorial on some of the chatbot's features. This way, the designer may prevent a communication breakdown when the user has to decide what to do after the first message.

  This strategy is used by TechCrunch and WSJ chatbots. On TechCrunch, while the first message of the chatbot is an **S1** strategy, the follow-up is a short tutorial about some of the chatbot's other features. These messages show the main menu, how to check the latest news, and how to subscribe to a topic of interest. So, on the first couple utterances, the chatbot is able to inform users about three distinct features and how to use them. WSJ is more discrete but still uses this strategy. After using **S1**, it uses *suggestions* to show its main features ("Latest news", "Trending topics", "Today's market", "Company news", "Help"). However, unlike TechCrunch, it is more subtle and does not show the main menu at this point.

- **S3 – Suggesting the next possible set of actions to the user.** Through this strategy, designers may avoid the situation in which the user does not know what to do after a response from the chatbot. This is achieved by sending the user *quick replies* or *cards* with buttons whenever the chatbot replies to users. These *quick replies* may be suggestions of what users can do next or a follow-up to the previous user request.

All of the chatbots use this strategy. CNN and TechCrunch use it when showing users the news stories, for example. *Carousels* with story *cards* always show a "more stories" button on the last *card* (TechCrunch) or a "something else" button on all *cards* (CNN), which will show users other related stories. Both chatbots use this strategy mostly when displaying the news and in a few other interactions. However, they usually lead the user into a straight path in the conversation. That is, they offer users more information about the current topic, but no alternative paths of conversation, such as a change of topic, or even the possibility to go back to a previous one. Some interactive paths will eventually lead to a dead-end, with no options for users to select.

WSJ uses this same strategy, but differently. It also shows a "more stories" button on the last *card* of a story's *carousel*. Nevertheless, it presents *quick replies* to the user on every utterance. That allows the user to easily access other topics or features instead of only following a straight path asking for more stories on the current topic.

It is interesting to note that, on all three chatbots, this strategy is not related to time constraints or delays of any kind. Instead, the *suggestions* of the next actions have a causal relation to the user's previous action, i.e, as soon as the user takes an action (sending a message or selecting an option, for example) the chabot already shows them the related *suggestions*.

- **S4 – Having a *persistent menu* with main features.** In this strategy, the designer chooses to create a *persistent menu*. This strategy may offer a solution to when a communication breakdown takes place. For instance, if the user forgets about how to access a feature, the *persistent menu* may be a way of accessing it. It also makes navigating through the chatbot easier on smartphones.

  All three inspected chatbots use this strategy. Although there are differences in what the designers choose to put in the menus, all of them include the main features of the chatbot.

  As mentioned before, on Messenger Web, the menu stays at the bottom left of the chatbot window. This menu stays closed until the user clicks on it[17], unveiling a list of the chatbot's features that may function as shortcuts so the user does not have to type a sentence.

  All the chatbots use this feature on Messenger App, i.e. the menu stays visible from the start, and the user has to select the option "Send a message" to be able to type a message to the chatbot.

[17] As of March 2017, Facebook made it possible for designers to hide the text-input box from the chatbots and show the *persistent menu* instead. So far, this is only possible on the smartphone Messenger App. This is further discussed on strategy **S10**.

- **S5 – Sending the main menu with main features as a message.**[18] This strategy is very similar to **S4**, except for the fact that the main menu is not visible all the time. Instead, this menu is a message displayed to the user as a reply and it remains in the chat history. For example, when the user says "menu" the chatbot replies the main menu itself. The main menu may comprise different sign classes, such as *quick replies* or a set of *cards* in a *carousel*. Usually, the main features are shown in the main menu so that users can easily choose what to do.

  This strategy was found in all three chatbots, but while the CNN menu is comprised of three *simple messages* followed by three *quick replies* to the main features. TechCrunch and WSJ (Fig. 4.11 on page 41) chatbots opt for a menu composed of a *carousel* with a few *cards* with buttons, each linking to a feature.

- **S6 – Having a list of available commands.** Another strategy for reminding users about the chatbot's features and how to use them. In this case, instead of a menu, the designer opts to present them a list of commands the chatbot can recognize. These commands are related to the chatbot's features. As in **S5** strategy, the command list is presented as a reply when the user asks for it.

  The only inspected chatbot that uses this strategy is the WSJ. As illustrated in Fig. 4.16, WSJ replies a list of commands to access its features when users select "command options" under the main menu. This reply is shown as a set of *simple messages* (text-only). Some of the features listed are also available on the *persistent menu*. Others are only mentioned in this list and the only way to use them is by typing the correct command.

- **S7 – Offering contextual help about a feature.** This strategy is used, for example, when a chatbot mentions a feature but does not explain how to use it. Then it offers help, in case the user does not know how to use it or how to trigger it. However, the offered help refers only to the previously mentioned feature and is not generic.

  TechCrunch uses this strategy. After the user sets a time to receive the daily digest, the chatbot informs that it is possible to change it under the subscriptions menu, and offers two *quick reply* answers: "Got it" and "Where's that?". If the user chooses the latter, the chatbot will respond in a few sentences where the menu is located (including *simple images* of screen captures with arrows pointing to it) and which functions can be found there, as shown in Fig. 4.17 on page 50.

[18] The name of the S5 strategy was changed from "Having a main menu with main features" ((Valério et al., 2016)) to "Sending the main menu with main features as message" to better express that the menu is displayed as a message to the user. But we have kept referring to the menu related to this message as "main menu" to differentiate from the S4's *persistent menu*.

Figure 4.16: WSJ's command list on mobile app. Source: Authors

- **S8 – Showing the main menu or the most frequent features when the user asks for help.** This strategy was identified on CNN and WSJ chatbots. It consists of showing the main menu or the most frequent features as response to a user's message asking for help. Usually, users ask for help when they do not know what to do next or when they do not know how to do something they want. Therefore, showing the menu (or features) is a good strategy to remind them of what the chatbot can do, and to explain how to do it.

  It is also worth noting that "help" is a command used in many command-line shells to obtain a list of all commands available. So a user that is used to a command-line interface is prone to make that connection and type "help" hoping to get a list of what the chatbot can do. It is also common to find a "help" section on many software, with instructions to guide the user through the interface and features.

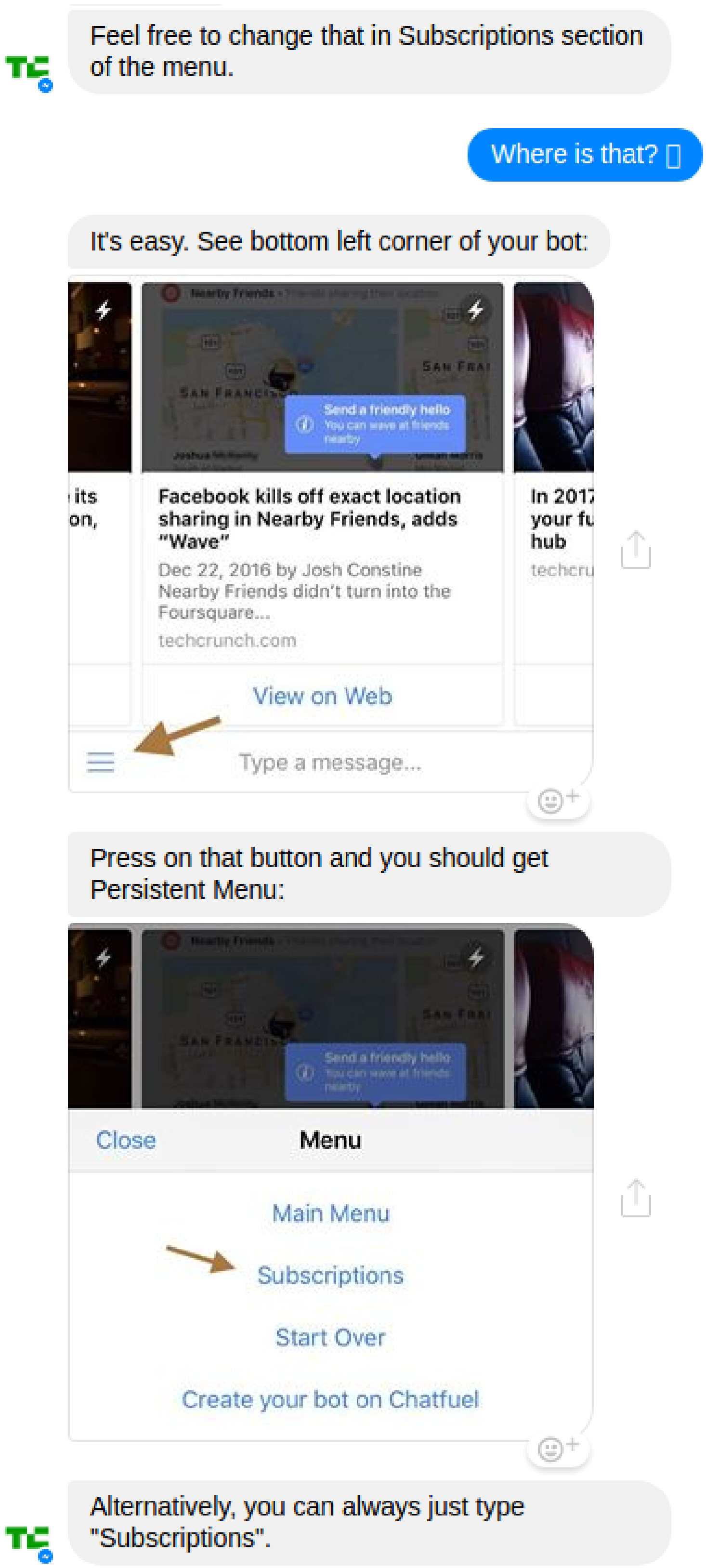

Figure 4.17: TechCrunch's contextual help about the subscriptions menu. Source: Authors

- **S9 – Showing the main menu or main features when user says something the chatbot cannot understand.** When the user types something the chatbot cannot understand, some of the reasons for it may include: (i) the user does not know how to access some feature of the chatbot; or (ii) they mistyped something. Both cases characterize communication breakdowns. This strategy offers a successful recovery from the breakdown caused by (i) and (ii) by refreshing the user's memory about what the chatbot can do and how to do so. Both CNN and WSJ chatbots adopt this strategy.

  When the user types something the chatbot cannot understand, all of the inspected chatbots will consider that the user refers to some content topic and the chatbot will try to search for news regarding what the user typed and show them. But only CNN and WSJ use the **S9** strategy as a follow-up.

  If no story is found, CNN chatbot replies saying it did not understand what the user wanted and shows a list of *suggestions* of what the user can ask it (Fig. 4.18). If the user keeps asking the same thing as before, after a few utterances, the chatbot will reply asking the user to try again or choose an option from a *card* containing the most used features. As for the WSJ chatbot, if no story is found, it will inform the user and will show the main menu *carousel* (Fig. 4.11 on page 41).

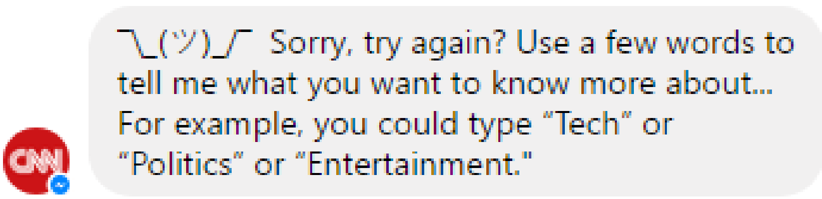

Figure 4.18: CNN's example of S9. Source: Authors

- **S10 – Showing the *persistent menu* instead of a text-input box.** At times, the user may not know or may not remember what the chatbot can do. Hence, instead of letting users type their own messages to discover what features the chatbot may offer, the designer may replace the traditional box for text inputs with the *persistent menu*. In this case, the *persistent menu* is visible all the time and the users must select an option to be able to type their own messages.

  This strategy has been made possible by an update to the Facebook Messenger platform[19] and it is only available on the Messenger app, and not on the web version. All of the inspected chatbots used this strategy when accessed through the smartphone

[19] https://messenger.fb.com/blog/messenger-platform-1-4-brings-even-more-tools-to-build-great-experiences, last access on Jan/2018.

app. As expected, this strategy was not available while chatting on the Messenger website.

This strategy may avoid communication breakdowns related to the user forgetting what the chatbot can do and how to access these features. But that comes at the cost of sacrificing some of the interface's conversational aspects. Nevertheless, it may be useful for chatbots with few features or with limited natural language processing capabilities.

- **S11 – Highlighting the most important features.** It is not unusual that a software has some features that are more important or more frequently used than others. In these cases, the most used features are usually highlighted on the interface, in a prominent place instead of under two or three layers of sub-menus. This strategy follows the same rationale on chatbots.

  Features the designer deems more important are highlighted through the combined use of other strategies, as showing the feature to the user on first messages (**S1**) and putting it in the *persistent menu* (**S4**) and the main menu (**S5**). On the other hand, features that are less used or less important are relegated to show up only on demand or in specific contexts.

  All of the inspected chatbots use this strategy. The most important features are easier to access, while the less important ones are kept out of sight. TechCrunch highlights the daily digest subscription, offering it on its first message (**S1**), in the main menu (**S5**), and in the *persistent menu* (**S4**). Nonetheless, choosing the time for sending the digest is a feature only shown when the user decides to check the subscriptions and chooses the "Setup Daily Digest" *quick reply.*

  The WSJ chatbot highlights the latest news feature, showing it on the first few messages (**S1**) and listing it in both persistent and main menus (**S4** and **S5**). An example of a less advertised feature is the companies comparison, which is not present in any menu of the chatbot. The user must type "compare" followed by the desired companies' stock ticker symbols to see a comparison chart. The command for comparing companies is only listed under the "What can I say" option in the main menu.

  The CNN chatbot highlights its "Editor's picks" feature, which is listed on the *persistent menu* (**S4**) and is also shown as an option on the main menu (**S5**). Additionally, there is a hidden feature: the "news stash". To access it, users must send a "thumbs up" emoticon to the chatbot and then select the "more" *quick reply.* Only then, users will be able to stash stories they like and check stories they have stashed.

* * *

In this section we presented the 11 strategies identified in the first round of inspections. The strategies were identified through the qualitative analysis using SIM, and not necessarily all of them were present in all three chatbots. Here is an overall view of how many of the chatbots adopt each strategy:

- Five strategies (45%) are used by all three chatbots – **S3** (*Suggesting the next possible set of actions to the user*), **S4** (*Having a persistent menu with main features*), **S5** (*Having a main menu with main features*), **S10** (*Showing the persistent menu instead of a text-input box*), and **S11** (*Highlighting the most important features*);
- Four strategies (36%) are used by only two of the chatbots – **S1** (*Showing the main feature on the first message*), **S2** (*Guiding the user through a short tutorial during first messages*), **S8** (*Showing the main menu or the most frequent features when user asks for help*), and **S9** (*Showing the main menu or the most frequent features when user asks for help*);
- Two strategies (18%) are used by only one of the chatbots – **S6** (*Having a list of available commands*) is used by WSJ; and **S7** (*Offering contextual help about a feature*) is used by TechCrunch;
- WSJ uses 10 of the 11 strategies (90,9%). **S7** (*Offering contextual help about a feature*) in the only strategy not used by it.

The full results of which strategies and sign classes were used by each chatbot can be seen on the three first columns (labeled TC, CNN, and WSJ) of Tables 4.1 and 4.2 on page 56.

4.7 Second Round Results: Findings Consolidation

This section shows the results of the second round of inspections that took place in January 2018 aiming to consolidate the strategies and sign classes that emerged during the SIM application. It is organized in two subsections, one that describes which sign classes and strategies were found in which chatbots and our conclusions about the indicators they raised; and the second in which we use sign classes and strategies to discuss specific design decisions in each of the chatbots.

4.7.1 Sign Classes and Strategies Presence in Other Chatbots

The results of our analysis are compiled on Tables 4.1 on the following page and 4.2 on page 56, which indicate the presence or not of each sign class or strategy, respectively, in each chatbot. The abbreviated chatbot names presented on each column on Tables 4.1, 4.2, and 4.3 are the following:

- TC (TechCrunch) – `https://www.messenger.com/t/techcrunch`
- CNN – `https://www.messenger.com/t/cnn`
- WSJ (Wall Street Journal) – `https://www.messenger.com/t/wsj`
- WP (Washington Post) – `https://www.messenger.com/t/washingtonpost`
- B9 (Brainstorm 9) – `https://www.messenger.com/t/brainstorm9`
- UOL (UOL Notícias) – `https://www.messenger.com/t/UOLNoticias`
- BOL – `https://www.messenger.com/t/BOL`
- PNC (Poncho) – `https://www.messenger.com/t/hiponcho`
- SMO (Smokey) – `https://www.messenger.com/t/smokeybot`
- BET (Beta) – `https://www.messenger.com/t/beta.feminista`
- DNK (Dankland) – `https://www.messenger.com/t/dankland`
- 18F (1-800-Flowers.com) – `https://www.messenger.com/t/1800flowers`
- 18FA (1-800-Flowers.com Assistant) – `https://www.messenger.com/t/1800FlowersAssistant`

It is important to note that TC, CNN, and WSJ were inspected in June 2017, hence the Tables reflect the state of those chatbots in that period[20]. The other chatbots were inspected in January 2018, and Poncho has been updated to bear a *persistent menu* during the inspections, and thus its results were updated.

Table 4.1 shows the results of the analyses of the six sign classes over the 13 examined chatbots (three from the initial SIM inspections plus ten from the consolidation). In the Table, a "✓" indicates that the sign class (rows) was present in the chatbot (columns), while a "." means otherwise. The columns are in the same order as the chatbots were

[20]The chatbots may have been updated since June 2017. In particular, we have noticed that CNN has undergone major changes.

analyzed, except for the last column, which shows the total of chatbots that used a particular sign class. Finally, the last row shows the total of classes used by a particular chatbot.

Table 4.1: Sign classes consolidation

	TC	CNN	WSJ	WP	B9	UOL	BOL	PNC	SMO	BET	DNK	18F	18FA	TOTAL
Simple message	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	13
Simple image	✓	.	.	.	✓	.	.	✓	✓	.	✓	.	✓	6
Quick Reply	✓	✓	✓	.	✓	✓	.	✓	✓	✓	.	.	✓	9
Card	✓	✓	✓	✓	✓	✓	.	✓	✓	✓	✓	✓	✓	12
Carousel	✓	✓	✓	✓	✓	✓	.	✓	✓	.	.	.	✓	9
Persistent Menu	✓	✓	✓	.	✓	.	.	✓	✓	.	✓	✓	✓	9
TOTAL	6	5	5	3	6	4	1	6	6	3	4	3	6	–

Through the second round of inspections we verified that, as expected, *Simple message* is the most used sign class, followed by *Cards*. The least used class is the *Simple image*, which was only used by six chatbots. The other sign classes (*Quick Reply*, *Carousel*, and *Persistent Menu*) were also commonly used in the inspected chatbots.

It is also interesting to notice BOL is the only chatbot to use just one sign class: the *simple message*, all the others use at least three of the sign classes in their communication. While BOL news messages mainly consist of the headline and link to a news story (see Figure 4.19 for an example), all the other news chatbots (WSJ, TC, CNN, WP, UOL, and B9) use a *card* when delivering news stories to the user, with a picture, the headline, a link to the story on their website and, sometimes, a small description (Fig. 4.10 on page 40 shows an example of a *card* presenting a piece of news on TechCrunch).

Also, Table 4.1 shows that 4 out of the 10 chatbots inspected in the second round use all of the sign classes, as does TechCrunch, inspected during the first round. Finally, during the second round of inspections, we did not identify any signs used by the chatbots that would represent a new sign class.

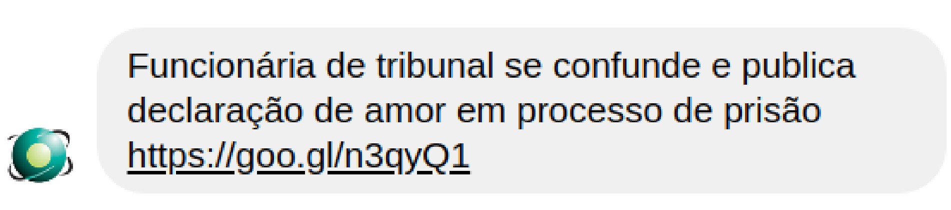

Figure 4.19: BOL chatbot's *simple message* with a news story. Source: Authors
Translation: *Court employee publishes love declaration in arrest sentence by mistake* `https://goo.gl/n3qyQ1`.

Regarding the strategies for conveying features, Table 4.2 on the next page compiles the results of the inspections looking for evidence of the strategies on the chatbots. As in Table 4.1, the rows represent the strategies (**S1** through **S11**), the columns show the

Table 4.2: Strategies consolidation

	TC	CNN	WSJ	WP	B9	UOL	BOL	PNC	SMO	BET	DNK	18F	18FA	TOTAL
S1	✓	.	✓	✓	✓	✓	✓	✓	.	.	.	✓	✓	9
S2	✓	.	✓	.	✓	.	.	.	.	.	.	.	.	3
S3	✓	✓	✓	.	✓	✓	.	✓	✓	✓	.	✓	✓	10
S4	✓	✓	✓	.	.	.	.	.	✓	.	✓	✓	✓	7
S5	✓	✓	✓	✓	✓	✓	.	✓	✓	.	.	.	✓	9
S6	.	.	✓	✓	.	.	✓	✓	.	.	.	.	.	4
S7	✓	.	.	.	.	.	.	✓	.	.	.	.	.	2
S8	.	✓	✓	✓	✓	✓	.	✓	✓	.	✓	.	.	8
S9	.	✓	✓	✓	.	✓	.	.	.	.	.	.	✓	5
S10	✓	✓	✓	.	✓	.	.	✓	✓	.	✓	✓	✓	9
S11	✓	✓	✓	.	✓	.	.	✓	✓	.	.	.	✓	7
TOTAL	8	7	10	5	7	5	2	8	6	1	3	4	7	–

S1 (*Showing the main feature on the first message*)
S2 (*Guiding the user through a short tutorial during first messages*)
S3 (*Suggesting the next possible set of actions to the user*)
S4 (*Having a persistent menu with main features*)
S5 (*Having a main menu with main features*)
S6 (*Having a list of available commands*)
S7 (*Offering contextual help about a feature*)
S8 (*Showing the main menu or the most frequent features when user asks for help*)
S9 (*Showing the main menu or the most frequent features when user asks for help*)
S10 (*Showing the persistent menu instead of a text-input box*)
S11 (*Highlighting the most important features*)

chatbots in the order the inspections took place, the last column shows how many chatbots used that particular strategy, and the last row informs how many strategies a particular chatbot has used.

It is interesting to note that every strategy was used by at least one chatbot considered in the second phase of our research. The least used strategies were **S7** (*Offering contextual help about a feature*), used only by two chatbots (CNN and PNC), and **S2** (*Guiding the user through a short tutorial during first messages*), which was used by three chatbots (TechCrunch, WSJ, and B9).

No strategy was used by all the chatbots. Nonetheless, some were much more popular than others. **S3** (*Suggesting the next possible set of actions to the user*) was used by 10 out of the 13 chatbots; while nine chatbots adopted **S1** (*Showing the main feature on the first message*), **S5** (*Having a main menu with main features*), and **S10** (*Showing the persistent menu instead of a text-input box*). Finally, **S8** (*Showing the main menu or the most frequent features when user asks for help*) was found in eight chatbots.

Initially, in our proposal of the strategies (Valério et al., 2017) we defined **S8** as "Showing the main menu or the most frequent features when user **says 'help'**". However, during the consolidation analysis, it came to our attention that many utterances led the chatbots to present to users their most frequent features or metalinguistic signs to help them with their interaction. Some of these utterances were explicitly interpreted by the chatbot as a request for help, while others had the same effect because the chatbot could not understand what the user meant – which was evidence of their adoption of **S9** (*Showing the main menu or the most frequent features when user asks for help*). These different utterances treated by the chatbot as a synonym for "help" led us to change the strategy **S8** description from "says help" to "asks for help".

Table 4.3 lists a set of sentences we identified as causing chatbots to present information to help the user move forward in the interaction. Each row indicates a different sentence/word, while the columns represent the chatbots, except for the last column and row that are the totals. The sentences chosen were some of those we considered users might utter when they want to ask the chatbot for help. In each cell, a "✓" indicates that the chatbot replied with a message stating (some of) its features. A "." marks otherwise. In some cases, the chatbot did not understand the sentence but responded with the features anyway. These cases are indicated with a "✓*" and are evidence of the adoption of **S9** (*Showing the main menu or the most frequent features when user asks for help*) by the chatbot.

Table 4.3: Features on Sentences

	TC	**CNN**	**WSJ**	**WP**	**B9**	**UOL**	**BOL**	**PNC**	**SMO**	**BET**	**DNK**	**18F**	**18FA**	**TOTAL**
"Help"	.	✓	✓	✓	✓	✓*	.	✓	✓	.	✓	.	.	8
"Menu"	✓	✓	✓	✓*	.	✓*	.	✓	.	.	.	.	✓*	7
"About"	.	.	✓*	✓*	.	✓*	.	.	.	.	.	.	✓*	4
"Hi", "Hey", or "Hello"	.	.	✓	✓*	.	✓	✓	.	✓	.	✓	.	✓	7
"What can you do?"	.	✓*	✓*	✓*	.	✓*	.	✓	.	✓	.	.	✓	7
"Commands"	.	✓*	.	✓*	.	✓*	.	.	.	.	.	.	✓*	4
"Features"	.	✓*	.	✓*	.	✓*	.	.	.	.	.	.	✓*	4
"What can I say?"	.	.	.	✓*	.	✓*	.	.	.	.	.	.	✓*	3
TOTAL	1	5	5	8	1	8	1	3	2	1	2	0	7	–

Note that the first five rows represent sentences or words that at least one chatbot understands as being a call for help. Interestingly, some chatbots (UOL, BOL, SMO, DNK, and 18FA) will respond to a greeting with the presentation of its main features. This could be interpreted as the chatbot "introducing itself", when someone greets it[21].

It is also worth noting that some other choices of sentences/words used in the analysis were based on the chatbots' messages sent to users. However, in some cases, the chatbot would use an expression when talking to users and it would not understand the same expression when said by the user. For instance, WSJ offered users a *card* with the button "What can I say?" (see Fig. 4.11 on page 41), but would not understand that same sentence if the user typed it.

And last, but not least, during the second round of inspections, we also identified a possible new strategy that was not present on any of the chatbots inspected on the first round. That new strategy was found on Poncho chatbot. Some days after the inspection,

[21] In the case of the greetings and the "About" there are chatbots that interpret it not as a call for help and do not reply with their features. So, for those cases, in this analysis, their cells are marked with "." (e.g., the TechCrunch chatbot).

Poncho sent us a message offering us to take a look at our horoscope. Prior to that, we did not subscribe to any kind of horoscope service in Poncho, so we concluded that Poncho was actively taking the initiative to offer one of its features. Fig. 4.20 shows the message Poncho sent us to offer the horoscope. Thus, this could be evidence of a strategy in which Poncho actively announces new features to users. However, a more extensive analysis of Poncho's dynamic signs would be necessary in order to be able to better understand in which conditions the chatbot issues such messages.

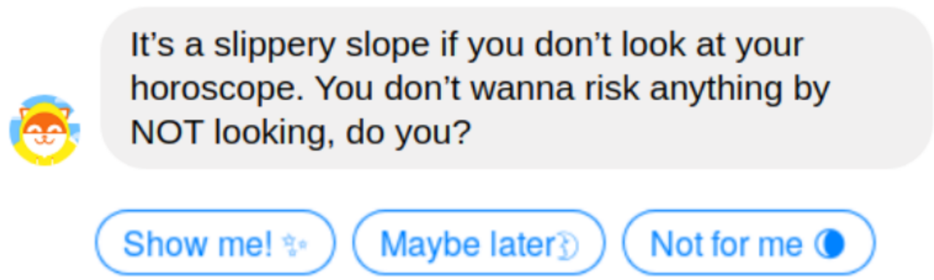

Figure 4.20: Poncho chatbot actively offering a feature. Source: Authors

4.7.2 Using Sign Classes and Strategies to Discuss Designers' Choices

While inspecting the *UOL Notícias* chatbot, we noticed how the designer's poor choice of the class sign to represent a communicative intent may result in potential communication breakdowns.

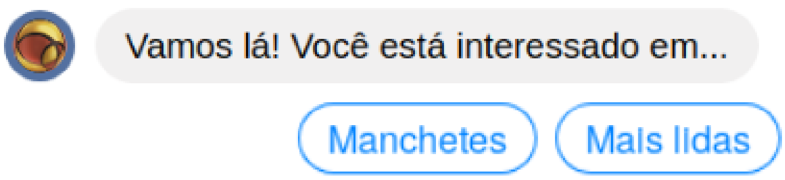

Figure 4.21: *Suggestions* on the *UOL Notícias* chatbot. Source: Authors
Translation: *–Let's go! You are interested in...*
-Headlines -Most read

Fig. 4.21 illustrates that: on its first message, the chatbot asks what the user is interested in and offers two *suggestions*: "Manchetes" (headlines) and "Mais lidas" (most read). When the user selects one of them, the *suggestions* disappear from the chat history, and the chatbot follows up asking the user at what time they want to receive the news digest. Only after that, does the chatbot inform the user that they can also choose to subscribe to the other option (in this case, "Mais lidas" – "Most read"), as shown on Fig 4.22 on the following page. When presented with the *quick replies* as in Fig. 4.21 on the previous page, a plausible interpretation would be that it is only possible to select one of the two options for the daily digest: either headlines or most read. However, later the chatbot goes back to ask users about the second option and informs them that if they would also like to subscribe to the other choice (no longer visible to them) they should say to the chatbot "receber" (receive). This communication could have been simpler by either adding a third *suggestion* for "both" in the set of *quick replies*, making them alternative choices; or by using a *card* that would allow users to click on the other option (at any moment) if they decided to subscribe to it.

As discussed in Section 4.6.3 - "Strategies For Conveying Features", while explaining **S1** (*Showing the main feature on the first message*), subscriptions to daily messages can help create a more lasting bond, because even if the user forgets about the chatbot, it will remind the user with a message every day. That may be a strategy for creating engagement with the user, but as this work focuses on strategies for conveying features, this will not be further discussed. Out of the chatbots that focus on news stories, almost all of them offer a subscription feature to users, the only exception is the Washington Post chatbot. WP does not offer any kind of subscription feature, instead relies on the user to ask for the "Top stories".

Another interesting fact about the Washington Post chatbot is that it can only understand three commands: "Top stories", "Contact", and "Help". If the user sends any message different from those three, the chatbot replies a message stating it can only respond to those three commands (Figure 4.23 on the following page shows that message). While that is a clear example of **S9** (*Showing the main menu or the most frequent features when user asks for help*), as the message comprises all the commands the chatbot understands, it is also a case of **S6** (*Having a list of available commands*); and, as the chatbot cannot understand a "help" message from the user, it also replies that same message (Figure 4.23 on the next page), which characterizes a case of **S8** (*Showing the main menu or the most frequent features when user asks for help*). That way, the Washington Post chatbot, despite being simple (only understands three commands), is able to follow three strategies to avoid and mend possible communication breakdowns.

Beta chatbot was found to use only one strategy: **S3** (*Suggesting the next possible set of actions to the user*). Beta is structured as a scripted conversation, the chatbot sends a message that ends with a question and it shows two *quick replies* as possible answers. Depending on which answer the users chooses, the next reply from Beta will be different.

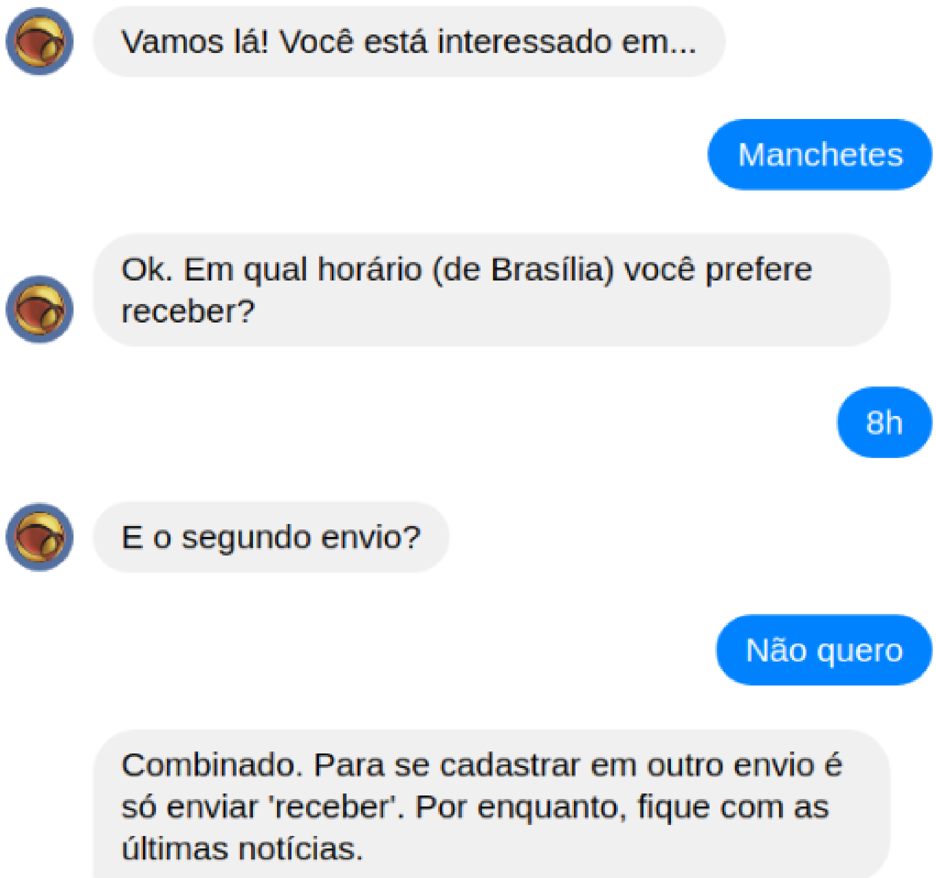

Figure 4.22: Follow-up on the *suggestions* on the *UOL Notícias* chatbot. Source: Authors
Translation: *–Let's go! You are interested in...*
–Headlines
–Ok. At what time (Brasilia time zone) do you want to receive it?
–8h
–What about second delivery?
–I do not want it
–Deal. For subscribing to the other delivery you just have to send 'receive'. For now, enjoy the latest news.

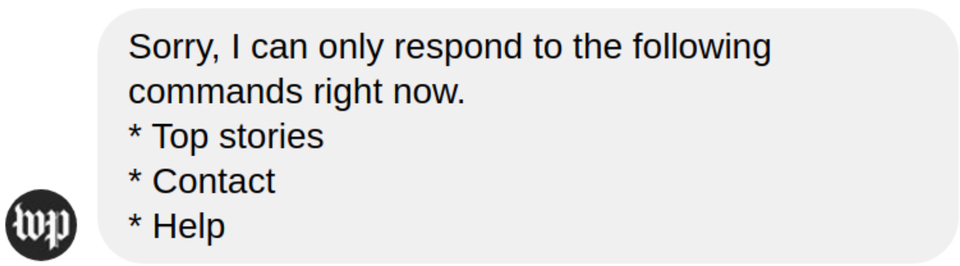

Figure 4.23: Washington Post chatbot's reply when it cannot understand the user's utterance. Source: Authors

However, if users try to type their own answer (or even typing the exact same text from the *quick reply*), Beta will say it cannot understand it and will show a *quick reply* for restarting the whole conversation. The only exception is when the user asks about what it can do, which is properly replied with a *simple message* stating what Beta can do. Other than that, there are no menus or help messages, just a conversation that must be followed through predetermined answers. During that conversation, Beta will inform the user about its purposes and about what it can do. It is a different approach from the other chatbots, as Beta has only one feature (send updates about feminist matters in Brazil), its designers opted for a (rather lengthy) conversation when presenting that feature.

The 1-800-Flowers.com chatbot is focused on customer service (in fact that is its only feature). During our inspections, we found a practice that had not yet appeared in any other chatbot: when the user selects the customer service option and writes a message to 18F, after a while, a real person will take over and answer the user in name of the chatbot. While it is a great way of ensuring that the user will not be frustrated by limited (or bad) Natural Language Processing, we were very surprised about it. People answering through the chatbot (presumably workers at *1-800-Flowers.com*) identify themselves by signing their messages, as it can be seen on Fig. 4.24 on the following page, in which "Sinead" answered an inquiry.

Another interesting point is that when inspecting the Brainstorm 9 chatbot, we noticed that some of its messages were familiar to us (as if we had heard them before). Brainstorm 9 is a Portuguese-speaking chatbot and we identified that its messages were direct translations from some of TechCrunch's utterances. An example of that is when the user sends the message "humor" to the chatbots, and both reply "Oh oh!". Other than that, the type of *cards* both chatbots used for displaying their news stories was the same: an image, the headline, and the button stating "View on Web" (Figure 4.10 on page 40) or its direct translation to Portuguese: "Ver na web", in the Brainstorm 9 chatbot. On further analysis, both chatbots display messages stating that they were made using the same platform: Chatfuel[22]. So it is possible that some of the replies were already coded in a template from that platform. That raises the question of which designer is saying what, as we can identify at least three of them in this case: the chatbot designer, of course, but also the Facebook Messenger designer, and the chatbot development platform designer. Thus, as our analysis indicates, the final chatbot language may include parts of the discourse of the three designers.

[22]`https://chatfuel.com`, last access on Jun/2020.

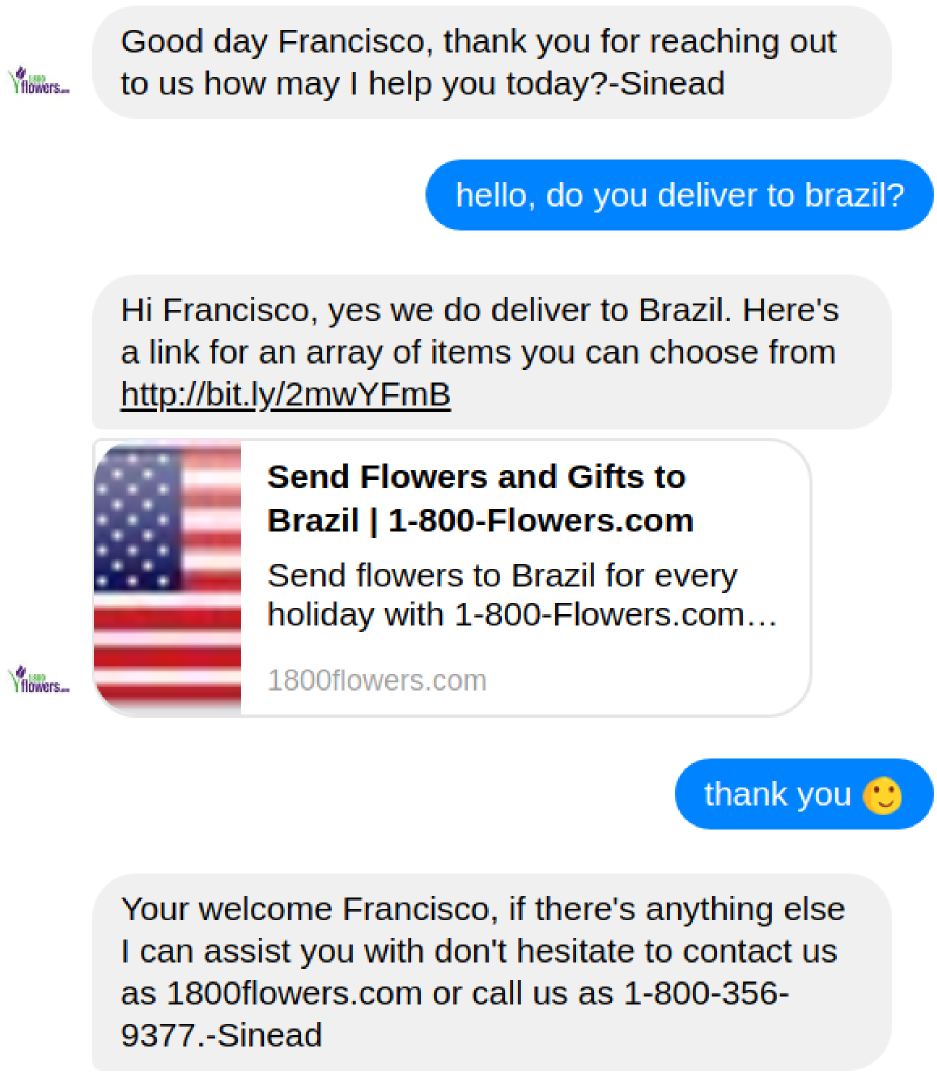

Figure 4.24: Human beings sending messages thorough *1-800-Flowers.com* chatbot. Source: Authors

4.8 Discussion

In the previous section, we discussed how the identified sign classes and strategies were used in a set of different chatbots, as well as how they could help designers or evaluators reflect upon their choices regarding the chatbot's interactive language. In this section, we go beyond the sign classes and strategies and discuss other challenges involved in designing chatbots' interactive languages, based on the problems they pose to the interaction identified in our analyses; considerations that could be helpful to researchers or professionals that would like to apply SIM to chatbots (or even other conversation-based interfaces); and reflections about possible threats to the validity of the present work.

4.8.1 Considerations About Chatbot Design

Some of the biggest challenges we noticed during the inspections of all chatbots were the openness of conversational interfaces interactive space and the hidden structure of chatbots. Thus, in this section, we discuss some considerations that could help designers tackle these issues.

An option for designers to deal with the openness of the conversational interface is telling the user what to expect from the system. For instance, relying more on metalinguistic signs for conveying how the user may interact with the interface. That way the user may focus on a reduced scope which might be more predictable. Hence, metalinguistic signs are even more important for grasping the intended metacommunication.

In this direction, some of the chatbots took the opportunity to "introduce themselves" when the user greeted them. Also, another possibility, as identified in our analysis, is including a tutorial about the features in the very first messages sent to the user (as in TechCrunch and WSJ). Another common strategy is to include responses to requests for help from the user, a help option on the *persistent menu*, or even to present users with metalinguistic signs anytime they could not interpret users' messages. Those "help" messages are usually composed of a list of main features, so the user may know what the chatbot can do and how to interact with it.

Another aspect that was noticed in our inspections was that often the user-to-chatbot (input) language is different from the chatbot-to-user (output) language. In some cases, the chatbots offered an option for the user that, if selected, would be shown on the chat history as if the user had typed it themselves, but the chatbot would not understand that same sentence when typed by the user. That was the case with TechCrunch (Figure 4.14 on page 44) and WSJ, which offered a *card* with the title "What can I say?" in its main menu (Figure 4.11 on page 41), but would not understand that same sentence when typed by the user (Table 4.3 on page 57, third column, second to last row). This can increase the cost of learning the interactive language for users since they cannot (completely) rely on the chatbot's discourse to learn how to communicate with it.

One approach that could make it easier for users to interact would be preventing them from composing the messages they send to the chatbot. This can be done by restricting the *simple message* and *simple image* sign classes to chatbot output only, and just offering users the possibility to choose from predefined options presented by the chatbot (by using menus, *quick replies*, or *cards*). This would be similar to bank systems that work through voice over a telephone. Although this solution would make it easier for users to interact with the chatbot, on the other hand, it would limit users' expressiveness while using the system, either by narrowing their vocabulary to a few sentences or by "putting words into their mouths" (as discussed on 4.6.2 - "Sign Classes Considerations").

In the same direction, an approach for addressing the hidden structure issue is the use of *cards* and *quick replies*. They help users navigate the chatbot, presenting possible actions to choose from, making it easier to see different possibilities users might not have thought of. But, if on one hand, the predetermined answers can help users to know the chatbot's features, on the other, it makes the interaction less like chatting and more like exploring a dialog tree.

The chatbot Beta mainly adopted this approach (as seen in subsection 4.7.2 - "Using Sign Classes and Strategies to Discuss Designers' Choices"), and although it allowed users to type messages, it only understood very few words, and most of the time it would present users with a default error message, with one or two possible *quick replies* that would lead to the continuation of the conversation. As discussed, chatbots allow users to type messages, but they encourage users to follow their predefined conversation scripts by offering options on how to continue most of the time or even making it more costly to write a message – as it is the case on the Facebook Messenger App in which users must select an option every time they would like to type a message (see Figure 4.12 on page 42 and Figure 4.15 on page 45).

The issues discussed in this section raise some of the challenges that come with designing chatbots. The sign classes and strategies identified in this paper can support designers in reflecting on the interactive language they offer users to interact with their chatbots. However, there are other relevant aspects to be considered, such as when and how to present metalinguistic discourse that will help users learn about the chatbot (without adding a large cost to the interaction), or the cohesiveness between the selected (from *cards* or *quick replies*) and typed messages the chatbot is able to interpret, and between the input language as a whole and output messages the chatbot sends to users.

4.8.2 Considerations About SIM Application to Chatbots

As presented in the section 4.1 - "Inspections Methodology", since SIM focuses on communicative aspects we could apply it to chatbots without needing to adapt or change its steps. Nonetheless, we presented the premises that we considered about how to classify a sign as metalinguistic, static, or dynamic in this context. In this section, we discuss some of the challenges we experienced when applying SIM in this research that can be useful to other researchers or professionals applying SIM to chatbots and even in other contexts.

First of all, SIM is a qualitative and interpretative method and, therefore, it relies on the specialist's own context, interpretation of the system's signs, and understanding of the system's intended user. These characteristics are emphasized when inspecting chatbots. Traditional graphical user interfaces (either mouse or touch-based) allow specialists to systematically explore all of the system's different options and menus, for they tend to be all exposed on the interface. On the other hand, Chatbots have their structure often hidden from the user. Chatbot's features and options stay hidden from the user until the right command is issued. It is a key characteristic of chatbots, and it comes at a cost.

This hidden structure makes it more difficult for specialists to explore the whole system, as they have to rely more on their own semiosis to choose which words to use and which tracks to take within the conversation. There are often more paths available on a conversation than in a traditional graphical user interface because the interactions are less restrained (users can say anything to the chatbot, there are countless possibilities – even if the chatbot does not understand all of them).

It may also be more difficult, for the inspector, to put themselves in the user's shoes because every person's semiosis is unique. While that is also true for traditional graphical user interfaces, when dealing with the openness of conversational interfaces it may become increasingly harder to emulate someone else's thoughts. The same can be said about the designers, for they also should anticipate how the users think as best as possible in order to design the interface.

The inspector must tackle the massive communication space available on the conversational interface. The first step for this is being aware of the amplitude of possible interactions and the difficulties to cover all the conversations that have been anticipated by the designer. The inspector should try out even uncanny possibilities, as it may reveal concealed features of the chatbot.

For that purpose, during the preparation step of SIM, it may be useful to create a list of usual (and unusual) sentences and actions for inspecting a chatbot. As a suggestion of generic sentences to be included on the list, one could use the ones shown on Table 4.3 on page 57 and their variations. The inspector may also define criteria of how to choose input messages to be evaluated. For instance, they could define a list of valid messages in other chatbots of the same domain or chatbots in general; or decide to include in the list all messages offered to users by chatbots as buttons or *quick replies* (i.e. does the chatbot understand the sentences/words it offers users to say when users type them?). That list may even be complemented and carried over several distinct inspections.

During the inspection, it is highly recommended that the inspector registers all input messages used or tried in the evaluation, as it represents the scope of the evaluation performed. In the cases in which there is more than one inspector, it will be helpful in their discussion to triangulate results. Furthermore, if more than one system is being investigated it will allow inspectors to systematically analyze the same scope in each one of them.

Finally, although SIM can be carried out by a single specialist, in the context of conversational interfaces, it may be interesting to consider using more specialists when inspecting chatbots (even in SIM technical applications), as that would potentially allow for a larger area of the communication space to be explored. For instance, during our inspections, only one specialist tried sending a "thumbs up" to the chatbot, and that revealed features that would otherwise be missing from the inspection (who would guess there would be a hidden menu with new features waiting for a "thumbs up" to be shown in the CNN chatbot?). With more specialists, each with their own mindset, distinct ways of exploring the interface may arise, making it easier to tackle the openness of the conversational interface.

4.8.3 Threats to Validity

As previously stated in the methodology, during the first round of inspections, we took a bottom-up approach to the analysis. We applied SIM to three similar-purposed chatbots to find out how the chatbots' designers were presenting their features to users (our research question to SIM). From the analysis emerged the six sign classes and eleven strategies, as described in section 4.6 - "First Round Results: Identification of Sign Classes and Strategies in Chatbot Communication". We then proceeded to the second round of inspections, taking a top-down approach to consolidate the findings of the first inspections.

The top-down approach was executed by performing a systematic inspection of a larger set of chatbots based on SIM – i.e. analyzing metalinguistic, static, and dynamic signs, but not applying the method completely. In our analysis we were able to identify the use of all sign classes and strategies in the chatbots, consolidating our findings from our first round. However, there is a chance that, by not having applied the SIM in a bottom-up approach, and also by focusing on our initial findings, we may have missed (or failed to identify) other signs classes or strategies.

Nonetheless, it is important to point out that in the second round analysis we did find evidence of a new strategy regarding how chatbots introduce new features (see subsection 4.7.1 - "Sign Classes and Strategies Presence in Other Chatbots"). The fact that one potential new strategy emerged indicates that the inspector was open to new findings, but does not mean that if a complete bottom-up approach had been applied other new strategies or sign classes would not have emerged. It is worth noting that this potential new strategy was in fact a new one (and not one that was present at the initial chatbots and was missed) that was related to an event (the release of a new feature) that had not been observed during the inspection of the other chatbots.

To fully consolidate the sign classes and strategies and minimize potential biases, the ideal would be to have different researchers applying SIM to a set of chatbots in different domains. They would take a bottom-up approach (as in our first round of analyses) and identify sign classes and strategies used in the analyzed chatbots and then triangulate their findings to the results presented in this work.

4.9 Conclusions

Even though chatbots have been around for a long time, few works are supporting their designers when making important design decisions. This work is a step in this direction, focusing on decisions about how to convey chatbots' features to users. Conveying the system's features to users is important since it might determine the system's success. This is especially difficult for text-based interfaces with features not immediately exposed to the user, but conveyed little by little.

This study is divided into two rounds of inspections. On the first, we used the scientific application of SIM on three popular news chatbots to find out what communicative strategies their designers used to inform users about their features. To the best of our knowledge, this was the first time SIM was used for analyzing conversational interfaces.

Two specialists inspected the chatbots, and the results were triangulated to consolidate the findings. Using these findings, we were able to answer our research question – "What communicative strategies have been used by popular chatbots to convey their features to users?". Our results show that designers of the analyzed chatbots use several communicative strategies. Overall, we identified six sign classes and 11 strategies associated with the chatbot interactive language design.

In the second round, we consolidate these strategies by analyzing other ten chatbots from various domains and two languages: English and Portuguese. The strategies were consolidated: every strategy was used by at least one chatbot on the second round. Besides, we found evidence of a potential new strategy: actively offering a feature to users. As mentioned in section 4.7.2 - "Using Sign Classes and Strategies to Discuss Designers' Choices", more extensive analyses are necessary to be able to better understand this potential strategy and consolidate it as an addition to our set of strategies. The sign classes were also consolidated, and no evidence of any new sign classes was identified.

Although all chatbots make use of the sign classes and strategies identified, each of the inspected chatbots shows a singular approach to its design, meaning their designers combine sign classes and strategies in a unique way to convey their intended metamessage. Thus, by explicitly identifying sign classes and strategies and discussing how they can be useful in dealing with the openness of the conversational space and supporting users' interaction with the chatbot, our work can be useful to both researchers interested in conversational interfaces and chatbot designers.

To further consolidate the strategies and sign classes discussed in this work, it would be interesting to perform new inspections on other chatbots. All of the 13 inspected chatbots used Facebook Messenger as the delivery platform. Other platforms (such as Telegram, Skype, or Kik) may have distinct visual representations for the sign classes we have identified or different classes altogether.

Furthermore, the strategies introduced in this work focus on introducing features to users; other strategies may be used for other ends, for example: dealing with communications breakdown, user onboarding, or convincing users to sign up to services. Once enough inspections are made and, consequently, the strategies and sign classes are satisfactorily mature, new guidelines and interaction patterns for designing chatbot interfaces can be derived. The present work is a step in that direction.

In this work, we inspected the final chatbot discourse presented to users. However, in our analysis, we found evidence that the final metacommunication is, in fact, a product written by multiple authors, or in the very least influenced or constrained by the different authors involved – the chatbot designer, the development platform designer, and delivery platform designer. During our second round of inspections, we came across some communicative acts from different chatbots that seemed identical, namely in Brainstorm 9 and TechCrunch chatbots. Looking further into the issue, we identified that both chatbots had been developed using the same platform (Chatfuel), as discussed in subsection 4.7.2 - "Using Sign Classes and Strategies to Discuss Designers' Choices". That may indicate that the chatbots' designers may have "inherited" some of the chatbots' discourse from the development platform.

Therefore, as future work, it could be interesting to analyze separately each one of these platforms and identify how much "say" each of these authors actually has in the final chatbot communication and interaction language; and if any of them create constraints or requirements regarding the class signs and strategies identified in this work.

Regarding the applicability of SIM on chatbots, our findings show that no modifications are needed to the method. Nevertheless, some issues should be taken into account – such as the classification of metalinguistic, static, and dynamic signs; and the challenges of exploring an open communication space. Regarding the former, metalinguistic and static signs make use of the same signification system, and a semantic and contextual analysis of the sign is necessary to classify it as metalinguistic or static. For the latter, having more than one inspector could be useful – even when not necessary for triangulation purposes (e.g. during a technical application or when triangulating with other compatible methods or theories) – as well as compiling a list of sentences and actions to be carried over distinct inspections.

In short, our analysis contributes to chatbot research, as it identifies strategies used by chatbots' designers to convey their features to users. It is also a step towards supporting these designers in deciding which strategies to use. Moreover, the strategies identified in these chatbots can be compared to strategies found on other types of chatbots and pave the way for creating a model that classifies these strategies; which would be useful for system designers. Also, the analysis of other platforms may contribute to the consolidation of the identified sign classes, also providing useful resources to designers and platform developers.

Finally, we also contribute to HCI knowledge by showing that our methodology (and therefore SIM) can be used to generate more knowledge about chatbots.

Chapter 5

Pragmatics and Chatbots

In this chapter, we continue our work by carefully classifying chatbots' messages under the light of some Pragmatics works in order to gain insight into the findings of the inspections from Chapter 4 - "Communicative Strategies Investigation".

Pragmatics is a field of Linguistics that studies how context influences meaning. We considered that it could help us better understand the communication happening with the chatbots, leading us to a better account of why certain strategies are more or less effective in this context. For that, we selected the three chatbots used on the first round of inspections from the previous chapter: TechCrunch, CNN, and The Wall Street Journal chatbots. These chatbots were chosen for the same reason as in Chapter 4: they have similar context, i.e. news, and their success may indicate that their designers made overall good decisions when creating them. Besides that, as we had already applied SIM to them, we had an initial analysis of their communication, and we also have records of their sentences (through the signs evidence necessary for SIM).

First, we present the works on Pragmatics we used when classifying chatbots' utterances on Section 5.1 - "Pragmatics", including Searle's Speech Acts (Subsection 5.1.1 - "Speech Acts"), Grice's Cooperative Principle (Subsection 5.1.2 - "Cooperative Principle"), and Leech's Politeness Principle (Subsection 5.1.3 - "Leech's Politeness Maxims"). Next, in Section 5.2 - "Chatbots Analysis", we show our analysis of the chatbots utterances. Including how we have adapted these works for analyzing chatbots on Subsection 5.2.1 - "Communicative Premisses". Then we present some examples of that classification (Subsection 5.2.2 - "Classification Examples"). Finally, we show our insights about the analysis on Subsection 5.2.3 - "Insights from the Analysis", our considerations on Section 5.3 - "Considerations", and our conclusions for this Chapter on Section 5.4 - "Conclusions".

5.1 Pragmatics

After the first results of using SIM on chatbots, we decided it could be fruitful to have an in-depth view of the chatbot speech, and that could lead to significant insights and contributions to chatbot design and evaluation.

That way we decided to inspect the utterances from chatbots under the prisms of Speech Acts (Searle, 1969), Cooperative Principle (Grice, 1975), and Politeness Principle (Leech, 1983, 2014). Next, we present these works.

5.1.1 Speech Acts

People use verbal communication not only to describe the world but also for having an effect on it. That is, people can use language to change (something in) the world (Austin, 1962). The Speech Acts theory makes a distinction between *locution*, *illocution*, and *perlocution*. When someone utters a sentence, it is possible to differentiate the words they used (*locution*), the intent (*illocution*), and the effects the speech has on the world (*perlocution*). Austin (1962) gives two examples for explaining that distinction:

> (E.1)
> Act (A)[1] or Locution:
> He said to me 'Shoot her !' meaning by 'shoot' shoot and referring by 'her' to her.
>
> Act (B) or Illocution:
> He urged (or advised, ordered, etc.) me to shoot her.
>
> Act (C. a) or Perlocution:
> He persuaded me to shoot her.
>
> Act (C. b):
> He got me to (or made me, etc.) shoot her.
>
> (E.2)
> Act (A) or Locution:
> He said to me, 'You can't do that'.

[1] In this quote, act A is an example of Locution, while B is a case of Illocution and C represents Perlocution.

Act (B) or Illocution:
He protested against my doing it.

Act (C. a) or Perlocution:
He pulled me up, checked me.

Act (C. b):
He stopped me, he brought me to my senses, etc.
He annoyed me. (Austin, 1962, pp. 102-103)

Following Austin's work, Searle (1969, 1975) proposed a taxonomy of five basic types of illocutions (or Speech Acts), according to how they may affect the world. The illocutionary Speech Acts are classified as:

- *assertives* (also called *representatives*) are the acts that commit the speaker to the truth of what is being said. Examples: stating, suggesting, claiming, announcing, predicting.
- *directives* are the acts in which the speaker tries to convince the hearer to do something. Examples: ordering, requesting, advising, begging, recommending.
- *declaratives* are the acts that change the status or condition of the world by virtue of what is being said, by whom, and to whom. Examples: resigning, dismissing, naming, sentencing, appointing.
- *commissives* are the acts in which the speaker commits themselves to do something in the future. Examples: promising, offering, vowing, undertaking.
- *expressives* are the acts in which the speaker expresses their psychological state or attitude. Examples: complimenting, accusing, congratulating, thanking, apologizing.

The Speech Acts taxonomy serves as the basis for other works on Pragmatics we also used when analyzing chatbots. These works are explained next.

5.1.2 Cooperative Principle

While communicating, the speaker and the hearer tend to cooperate in order to convey to the hearer what the speaker meant. Grice (1975) says that, when people communicate, depending on the stage of the conversation, some reactions are suitable, while others are unsuitable, for a proper dialogue. That way, Grice formulates the Cooperative Principle (CP): "Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged" (Grice, 1975, p. 45). Although it may sound prescriptive, it is intended to be a description of how people usually communicate, that is: people are expected to observe the CP while communicating.

Grice breaks the Cooperative Principle into maxims, which are divided into four categories: (Grice, 1975)

- Maxim of *quantity*, related to the quantity of information provided by the speaker.
  - Speakers should make their contribution as informative as required (for the purposes of the ongoing exchange).
  - Speakers *should not* make their contribution *more* informative than required.
- Maxim of *quality*, related to the truthfulness of the information provided by the speaker. "Try to make your contribution one that is true." (Grice, 1975, p. 46)
  - Speakers *should not* say what they believe to be false.
  - Speakers *should not* say anything for which they lack adequate evidence.
- Maxim of *relation*, related to maintaining the topic of a conversation.
  - Speakers should make relevant contributions.
- Maxim of *manner*, related to the way the information is being conveyed. "Be perspicuous." (Grice, 1975, p. 46)
  - Speakers *should avoid* obscurity of expression in their contributions.
  - Speakers *should avoid* ambiguity in their contributions.
  - Speakers should be brief (avoid unnecessary prolixity) in their contributions.
  - Speakers should make their contributions in an orderly manner.

Grice (1975) stresses the importance of the first maxim of quality ("Do not say what you believe to be false") and points that the other maxims only come into operation on the assumption that this quality maxim is fulfilled. Regarding the second maxim of quantity (over-information), it is pointed that, in not respecting it, the hearer may think there is a reason for the excess of information when there is not. Regarding the relation maxim, it is important to consider the *natural* change of topic during conversations.

Grice (1975) also explains how the Cooperative Principle can help to understand *conversational implicatures* that may exist in conversations. Conversational implicatures are cases of (hidden) messages that may be implicit in a conversation. By analyzing whether the speaker is fulfilling, violating, opting out of, or flouting the maxims it is possible to check for a concealed meaning in the message.

Grice (1975) exemplifies three different types of conversational implicatures caused by the observation (or not) of the Cooperative Principle:

1. **No maxim is (apparently) violated.**

   A is next to an immobilized car when B approaches:

   A: I am out of gas.

   B: There's a gas station around the corner.

   In this case, A must assume that B is following the Cooperative Principle, and the information B gives complies with the relation maxim. That way, A may assume that the gas station around the corner is open and has gas available.

2. **A maxim is violated to avoid clashing with another maxim.**

   A and B are planning a holiday trip to France, where their friend C lives. Both A and B know that A would want to see C if possible.

   A: Do you know where C lives?

   B: Somewhere in southern France.

   B knows that her answer is less informative than necessary (thus violating the quantity maxim). But that is done so B avoids violating the quality maxim by giving information she has no adequate evidence for (as B does not know where in southern France C lives).

3. **A maxim is exploited with the intent to convey additional meaning that was not literally expressed.**

   A is writing a recommendation letter for her pupil X for a philosophy job. The letter reads: 'Dear sir, Mr. X's command of English is excellent, and he has regularly attended tutorials. Yours, A'

   In this case, we must assume that A has not opted out of the Cooperative Principle, otherwise, she would have not written the letter in the first place. We cannot say that she has little information on X, since X is her pupil. And A knows that what is demanded is more information about X. A is flouting the quantity maxim (she is withholding information about X), therefore, the implicature is that she does not think X is good at philosophy.

Finally, Grice (1975) notes that the maxims of the Cooperative Principle are not the only ones that can be observed in communications, and other maxims can also be observed during conversations, such as "Be polite". Later, Leech (1983) would publish his Politeness Principle, complementing Grice's Cooperative Principle.

5.1.3 Leech's Politeness Maxims

Following Grice's work, Leech (1983) argues that the Cooperative Principle alone cannot explain why so often people are indirect when communicating. Thus Leech presents other maxims that are also associated with the communicative process. These maxims are related to the use of politeness during exchanges and are grouped into Politeness Principle (PP). To Leech (1983), polite behavior is what allows people to interact with each other in relative harmony. That way, the PP is what makes it possible for us to assume people are cooperating while communicating, or as Leech puts it: "unless you are polite to you neighbour, the channel of communication between you will break down, and you will no longer be able to borrow his mower" (Leech, 1983, p. 82). Therefore, when communicating, people tend to avoid communicative discord and seek communicative concord.

5.1.3.1 Politeness Scales

Leech (1983, 2014) shows two ways of analyzing the politeness phenomena, by using two different scales for interpreting how polite an exchange is:

- *Pragmalinguistic politeness scale* (formerly called "Absolute politeness scale" in (Leech, 1983)), in which we can order utterances in a unidirectional scale without considering the context. For example: "Thanks" is considered less polite than "Thank you very much" on this scale.
- *Sociopragmatic politeness scale* (formerly called "Relative politeness scale" in (Leech, 1983)) is the politeness relative to the context, that is, the society, group, and situation in which the sentence is uttered. This scale is bidirectional, and sentences can be classified as underpolite, overpolite, or having adequate politeness. For example, "thanks" may be adequate politeness in an informal conversation among friends, but it can be classified as underpolite when replying to a big favor your boss did for you.

In order to assess the appropriate degree of politeness on the sociopragmatic politeness scale, Leech (2014) borrows some concepts from Brown and Levinson (1978, 1987):

- *Vertical distance* between Self and Other (in terms of status, power, role, age, etc.), as Brown and Levinson (1978, 1987)'s P factor.

- *Horizontal distance* between Self and Other (are they intimate, familiar, acquaintance, stranger, etc.), as Brown and Levinson (1978, 1987)'s D factor.

- *Cost/benefit* of the favor/obligation being asked, as Brown and Levinson (1978, 1987)'s R factor. "How large is the benefit, the cost, the favor, the obligation, etc., or in other words, the real socially defined value of what is being transacted."(Leech, 2014, p. 103)

The need for politeness may change when distances are reduced. For example, when communicating with people you are intimate with, there is no need to be as polite as when talking to someone you do not know very well. Furthermore, excessive politeness can even be misinterpreted as irony or sarcasm in some cases (e.g., saying "Could I possibly interrupt?" to a family member that is monopolizing the conversation can be seen as sarcasm).

Leech (2014) points that these scales vary from culture to culture, and that what may be perceived as impolite in one culture can be the norm in another. Even the cost-benefit can differ depending on where you are: it can be easier to borrow umbrellas in Japan than in the United States.

5.1.3.2 Illocutionary and Social Goals

To Leech (1983, 2014), some illocutions (i.e., the intention associated with the speech) are inherently polite (such as offers) or impolite (the case of orders), he then distinguishes between positive politeness (pos-politeness) and negative politeness (neg-politeness). The first maximizes the politeness of polite illocutions (for example, when complimenting someone, as you put a high value to that person's qualities), while the latter consists in minimizing the impoliteness of impolite illocutions (for example, when making a request, you may want to avert offense by avoiding imposing your goals on another person).

A good clue for identifying pos-politeness is checking whether the degree of pragmalinguistic politeness of the utterance can be increased through the use of intensifying modifiers. For example: when thanking someone for a favor, one may say just "Thank you", but the sentence's pragmatic politeness degree can be increased by using an intensifying modifier such as "Thank you **very much**". That way we can infer that paying compliments is a case of pos-politeness. On the other hand, neg-politeness is usually made pragmalinguistically more polite through the use of hedges and downgrading. An example is when making a request, it would be more (pragmalinguistically) polite to say "**Would** you **mind just** being quiet **for a moment**?" than just saying "Be quiet".

Leech (2014) also considers Searle (1969, 1975)'s classification of Speech Acts very useful when analyzing politeness, as some of Searle's categories are associated with politeness: speech acts classified as directives are associated with neg-politeness, while commissives and expressives include illocutions usually related to pos-politeness.

According to Leech (1983, 2014), when communicating, we have illocutionary goals (the intention behind the speech, like asking a favor) and also social goals (related to maintaining good communicative relations to other people). Sometimes these two goals may compete with each other (e.g., when requesting to borrow a car, Self's illocutionary goal – to borrow the car – is competing with the social goal – keeping good relations with Other, as Self may be imposing on Other to borrow their car), and other times they may cooperate (e.g., when complimenting Other, Self's illocutionary goal – to express admiration towards Other – goes on the same direction as the social goal – by complimenting Other, Self will probably maintain their relation on good terms). Leech (1983) divides speech events according to their illocutionary function as:

- *Competitive*, when the illocutionary goal competes with the social goal. Subjected to neg-politeness, as the speaker tries to reconcile social and illocutionary goals. Examples: ordering, asking, demanding, begging, etc.
- *Convivial*, when the illocutionary goal coincides with the social goal. Subjected to pos-politeness. Examples: offering, inviting, greeting, thanking, congratulating, complimenting, etc.
- *Collaborative*, when the illocutionary goal is indifferent to the social goal. The participants have the same goals that do not compete nor contribute to the social goal, so they do not involve politeness. Examples: instructing, reporting, asserting, announcing, etc.
- *Conflictive*, when the illocutionary goal conflicts with the social goal. There is no reason to use politeness (except ironically) when trying to offend others. Examples: threatening, accusing, cursing, etc.

5.1.3.3 General Strategy of Politeness

Leech (1983) describes the Politeness Principle maxims in terms of Self (or speaker) and Other (or hearer). Each maxim of the Politeness Principle is accompanied by a submaxim of less importance. While the maxims are focused on minimizing neg-politeness, the sub-maxims maximize pos-politeness. Thus, minimizing neg-politeness is considered more important than maximizing pos-politeness. The maxims, sub-maxims (inside brackets), as well as the speech acts in which they appear are the following:

- *Tact* maxim: minimize cost to Other, [and maximize benefit to Other]. Used in directive and commissive speech acts.
- *Generosity* maxim: minimize benefit to Self, [and maximize cost to Self]. Used in directive and commissive speech acts.
- *Approbation* maxim: minimize dispraise of Other, [and maximize praise of Other]. Used in expressive and assertive speech acts.
- *Modesty* maxim: minimize praise of Self, [and maximize dispraise of Self]. Used in expressive and assertive speech acts.
- *Agreement* maxim: minimize disagreement between Self and Other, [and maximize agreement between Self and Other]. Used in assertive speech acts.
- *Sympathy* maxim: minimize antipathy between Self and Other, [and maximize sympathy between Self and Other]. Used in assertive speech acts.

In (Leech, 2014), the Politeness Principle's maxims are reformulated into a super-constraint (or supermaxim) called General Strategy of Politeness (GSP). As Leech puts it: "General Strategy of Politeness: In order to be polite, S expresses or implies meanings that associate a favorable value with what pertains to O [other, hearer] or associates an unfavorable value with what pertains to S (S = self, speaker)." (Leech, 2014, p. 90)

As Leech states, when using the GSP, one tries not to offend others, avoiding pushing their own agenda to the other party. In this reformulation, Leech also adds four more maxims to the Politeness Principle, although he claims that these maxims are just manifestations of the supermaxim, the GSP. The reformulated maxims are shown in Table 5.1.

On Table 5.1 on the following page, the odd-numbered maxims (indicated in the first column) make use of pos-politeness (taking away value from S), while the even-numbered ones use neg-politeness (giving value to O). The second column indicates the related pair of that maxim and the third column shows each maxim's label.

The fourth column of Table 5.1 (Typical speech event) shows the type of speech act that the maxim is usually associated with. *Commissives* and *directives* classifications are taken directly from Searle (1975)'s taxonomy, while the others (compliments, self-devaluation, apologizing, etc) are examples of *expressive* speech acts.

Table 5.1: GSP's maxims. Adapted from Leech (2014)

Maxim	Related pair	Label	Typical speech event	Typical orientation (S or O)
(M1) give a high value to O's wants	Generosity, Tact	Generosity	Commissives	S
(M2) give a low value to S's wants		Tact	Directives	O
(M3) give a high value to O's qualities	Approbation, Modesty	Approbation	Compliments	O
(M4) give a low value to S's qualities		Modesty	Self-devaluation	S
(M5) give a high value to S's obligation to O	Obligation	Obligation (of S to O)	Apologizing, thanking	S (apologizing) O (thanking)
(M6) give a low value to O's obligation to S		Obligation (of O to S)	Responses to apologies and thanks	O (resp. to apologies) S (resp. to thanks)
(M7) give a high value to O's opinions	Opinion	Agreement	Agreeing, disagreeing	O
(M8) give a low value to S's opinions		Opinion reticence	Giving opinions	S
(M9) give a high value to O's feelings	Feeling	Sympathy	Congratulating, commiserating	O
(M10) give a low value to S's feelings		Feeling reticence	Suppressing feelings	S

Finally, the last column of Table 5.1 shows whether that maxim is typically S or O-oriented. The maxims presented are related to speech events that involve some kind of transaction, in the sense that the S[peaker] or the O[ther] person involved will have to carry out a social action (e.g.: in an offer, S will have to perform a favor; on the other hand, in a request, O will have to do something). Thus, the "O-oriented maxims place a *higher weighting* on what pertains to O, and the S-oriented maxims place a *lower weighting* on what pertains to S" (Leech, 2014, p. 92) (this weighting should be seen as a numerical scale, in which the higher number outbids the lower one).

Leech (2014) also notes that some maxims can be seen as action-motivating (e.g., M1 – Generosity maxim) or, less commonly, as action-inhibiting (e.g., M10 – Feeling reticence). He also points that O-oriented maxims are usually more powerful (i.e., people tend to adhere more to the maxim) than the S-oriented ones, with the exception of Tact (M2) maxim being perceived as more powerful than Generosity (M1), at least in anglophone societies. Leech (2014) also notes that neg-politeness maxims are more powerful than pos-politeness ones and that the maxims higher up on Table 5.1 are more powerful than the ones on the bottom of the list, although Leech says that it can vary from culture to culture and needs further investigation.

Finally, the Politeness Principle (PP) Maxims are not supposed to be followed at all times like the Cooperative Principle (CP) ones, as people do not always want to be polite. Besides that, Leech (2014) points out that sometimes the PP maxims may clash among themselves, or even with the CP maxims. For example: when giving advice, the Generosity maxim can compete with Agreement and Modesty maxims (as S implies that their idea is more valuable than O's); or in case of excessive flattery, in which the PP maxim of Approbation may clash with the CP maxim of Quality (exaggerated to a point of not being truthful anymore).

5.2 Chatbots Analysis

This section details how we used the previously explained Pragmatics concepts while analyzing utterances from chatbots. Subsection 5.2.1 - "Communicative Premisses" shows these concepts and how they were used. Next, on subsection 5.2.2 -"Classification Examples" we explain the rationale behind the classifications through some examples.

5.2.1 Communicative Premisses

As we stated before, some aspects of the works of Searle, Grice, and Leech were used to help in the investigation of the analyzed chatbots and their utterances. To do so, each utterance was classified according to:

- its illocutionary speech acts (as Assertive, Directive, Declarative, Commissive, or Expressive), according to Searle (1969, 1975)'s taxonomy;
- whether or not it respects Grice (1975)'s Cooperative Principle's four maxims (Quality, Quantity, Relation, and Manner);
- whether or not it respects any of Leech (2014)'s 10 maxims from the Politeness Principle (Generosity, Tact, Approbation, Modesty, Obligation of S to O and O to S, Agreement, Opinion reticence, Sympathy, and Feeling reticence);

- its illocutionary and social function, according to Leech (1983)'s classification into Competitive, Convivial, Collaborative, and Conflictive;
- its pragmalinguistic and sociopragmatic politeness, according to (Leech, 2014).

Each utterance was also classified according to the strategies used to convey the chatbot's features (see Chapter 4 - "Communicative Strategies Investigation", subsection 4.6.3 - "Strategies For Conveying Features" for details about the strategies).

While the illocutionary function and evoked maxims are straightforward to use for classifying the chatbots' utterances, the pragmalinguistic and sociopragmatic politeness scales are not. Thus, the approach we took to assess the pragmalinguistic politeness was to consider a scale that goes from 0 (when there is no politeness involved – but *impoliteness* is not necessarily involved) to 5, which would be the case of an extremely polite sentence. The other values would mark the middle ground between these two. That way, the pragmalinguistic politeness scale is the following:

- 0: no politeness involved;
- 1: a little polite;
- 2: polite;
- 3: very polite;
- 4: greatly polite;
- 5: extremely polite.

An important caveat is that the author, who performed the analysis of the chatbot's sentences, although fluent in the English language, is not a native speaker. That may have impacted the classification process.

Regarding the sociopragmatic politeness scale, when talking to a chatbot, the user does not have a "real" relationship with it. That way we had to infer the vertical and horizontal distances between the user and the chatbot. To do it, we first analyze the overall way the chatbot speaks to the user, checking if the chatbot uses a more informal approach or not, in order to make an inference about the appropriated vertical and horizontal distances. This is similar to reading a novel in which two characters are having a conversation and trying to discover the relationship between these characters. After we have set the values for the horizontal and vertical distances, it is possible to check the weightiness (cost-benefit) of the transaction and classify the utterance in the sociopragmatic scale. We considered the cost associated with the user when grading the cost-benefit of the chatbot's messages.

That way, the possible values for vertical distance (differences in power) are the following:

- 0: both user and chabot are in the same level;
- 1: the distance between a clerk (lower level) helping a customer (higher level);
- 2: the distance between an employee (lower level) to their boss (higher level);
- 3: the distance between a commoner (lower level) and an authority (higher level).

For the horizontal distance (familiarity) between the chatbot and the user, the possible values are the following:

- 0: intimate;
- 1: familiar;
- 2: acquaintance;
- 3: stranger.

Finally, the values for the sociopragmatic scale of politeness are listed next:

- -3 = extremely underpolite;
- -2 = very underpolite;
- -1 = underpolite;
- 0 = adequate politeness for the situation;
- +1 = overpolite;
- +2 = very overpolite;
- +3 = extremely overpolite.

That was done to check whether there is any relation between the analyzed dimensions, and also to gain new insight on how designers crafted the chatbot's discourse and the way it contrasts to the way humans make conversational exchanges.

5.2.2 Classification Examples

Using the previously mentioned method, we classified every message sent by the three chatbots during the first round of inspections using the Semiotic Inspection Method (as detailed in Chapter 4 - "Communicative Strategies Investigation"), as the interaction with the chatbots was recorded during SIM application. Although all of the chatbots' messages were classified, it is not practical to show all of the results in this work.

This subsection shows examples of how the chatbots' utterances were classified. It aims at illustrating the rationale behind the analysis and at better explaining the dimensions used in this work.

5.2.2.1 Example 1: CNN's News Story

This first example depicts an utterance from CNN chatbot. The user is checking the news stories and decides to see other news, so the user selects "Something else" from the news *card*. In response, the chatbot sends another story *card* (similar to the previous one, but with a different news story) preceded by the following: "Here's a story I thought you might like, based on what you've been reading". This sequence can be seen in Figure 5.1.

- **Strategy**: **S3** (*Suggesting the next possible set of actions to the user*). The use of the S3 strategy is evidenced by the suggestions on the news card ("Read this Story" and "Something else").

- **Sign Type**: Static. The message is not explaining the chatbot or its capacities, so it is not a case of a metalinguistic sign. That way, we classify it as a static sign. Note that while the message itself is classified as a static sign, when assessing other dimensions like the Cooperative Principle, for example, we need to consider the context of the message, that is, to which request it is responding; in that case, one could argue that the message is a dynamic sign that was sent by the chatbot as an answer to the previous message. But the message itself is a case of static sign.

- **Speech Act**: Directive. The starting point is the user themselves asking for another story (by replying "Something else" to a previous message), that way, the chatbot is answering to the initial demand, but it is doing so by recommending a few stories the user "might like". Because of that, the illocutory Speech Act is classified as directive.

  On the other hand, in each news *card*, the chatbot presents there are two buttons: "Read this Story" and "Something else". That would be the case of the chatbot's designer committing themselves to show the user "this Story" or "Something else" if the user desires so. By analyzing only the buttons in the *cards*, the utterance would be a case of commissive Speech Act, but as it is part of a larger context, we considered that the directive part of the message is more important than the commissive one, hence we classify it as directive.

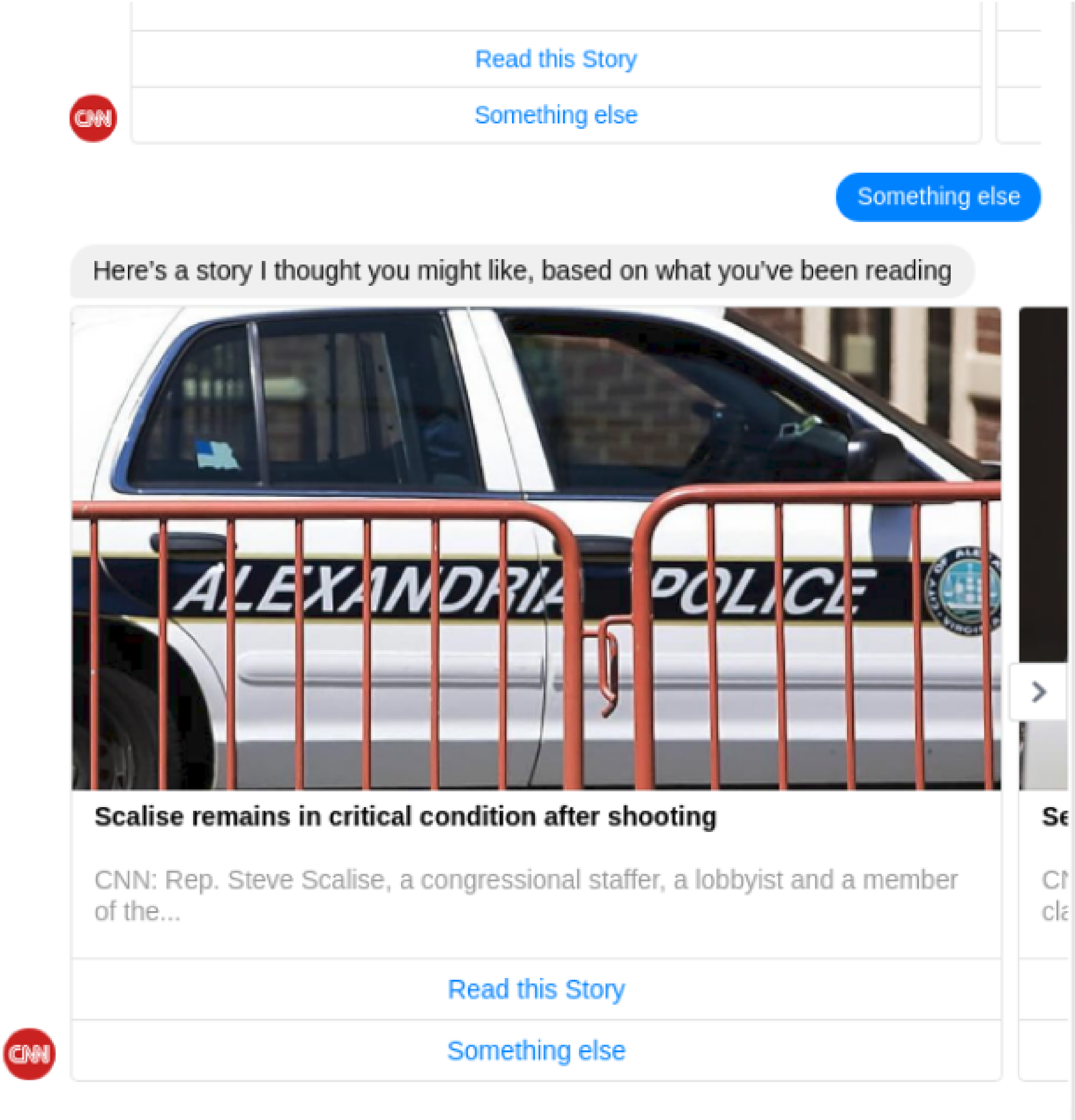

Figure 5.1: Example of utterance from the CNN chatbot. Source: Authors

- **Illocutionary function**: Convivial. In this example, both the user and the chatbot have a common goal: the user wants to read "Something else" and the chatbot was designed to provide its user with news stories, so, in a way, the chatbot is offering the news stories to the user, characterizing the convivial illocutionary function. The second part of the message (the "Read this Story" and "Something else" buttons) can also be classified as having a convivial illocutionary function, as the chatbot is offering to show the story or recommend something else.
- **Cooperative Principle**
  - **Quality**: Respected. The chatbot is presenting other stories, and we have

no reason to consider it is not true that they are “based on what you’ve been reading”.

- **Quantity**: Respected. The chatbot sends a group of four different stories to the user in a *card carousel*. It is not a whole lot of stories nor too few.
- **Relation**: Respected. The user asked for “Something else”, and the chatbot replied with other stories distinct from the previous ones.
- **Manner**: Respected. The stories are displayed in an orderly manner, as well as the sentence presenting the news stories (although it says it’s “a story” and presents the user with several stories).

- **Politeness Principle**
  - **Generosity**: Respected. Although the presented stories were asked by the user before, the chatbot also offers to fetch other stories through the “Something else” button, and that offer was not a response to a previous demand from the user. That way, we can consider the sentence as adhering to the Generosity maxim.
  - **Tact**: Not used. We did not find evidence that this maxim was evoked.
  - **Approbation**: Not used. We did not find evidence that this maxim was evoked.
  - **Modesty**: Respected. Even though the chatbot can consider the stories it is sending as adequate to the user, “based on what you’ve [the user] been reading”, the sentence uttered by the chatbot gives low value to its qualities (such as its ability to suggest stories that are adequate users). The use of hedging (“thought you might”) is also indicative of neg-politeness, which is the case for the modesty maxim.
  - We did not find any evidence of Maxims of **Obligation of S to O**, **Obligation of O to S**, **Agreement**, **Opinion reticence**, **Sympathy**, nor **Feeling reticence** in this message.

- **Pragmalinguistic politeness**: Very polite (3 out of 5). The sentence is very polite, especially the hedging part (“I thought you might like”), which could be rephrased as less polite versions, such as “you’ll like” or “you may like”.

- **Vertical distance (P)**: Clerk to a customer (1 out of 3). The values for vertical and horizontal distance are fixed throughout the messages from the chatbots. As explained before, they are defined after analyzing the way the chatbot forms its sentences overall. Therefore, all the messages from a particular chatbot (CNN, in this case) will be classified with the same vertical distance. The chatbot speaks as a salesperson that is trying to help a customer. So it is not on the same level as the customer (user) and we may consider that the customer is one level above the salesperson (chatbot).

- **Horizontal distance (D)**: Acquaintance (2 out of 3). The horizontal distance is also fixed throughout the chatbot's messages. The CNN chatbot behaves constantly as someone that is not familiar to the user, although not as a stranger. That can be evidenced by the lack of slang language and the use of hedging when asking something to the user. However, there are a few messages that betray that (un)familiarity, for example, the use of emoji when the chatbot cannot understand the user's request, (as seen on Figure 4.18 on page 51) and the hidden options that are shown when the user sends a thumbs-up (see Chapter 4 - "Communicative Strategies Investigation", subsection 4.6.3 - "Strategies For Conveying Features" on page 52). Even so, the chatbot overall talks like an acquaintance to the user, hence the classification as 2.

- **Cost-benefit (R)**: 1 (out of 5). Usually, sending the requested news stories would not be considered as something heavy for the user, for the user does not have to do anything. But there is a catch: the chatbot is also informing that the news stories it is sending are tailored for the user based on "what you've [the user] been reading". That is: the chatbot is admitting to be monitoring the user's actions, which could be considered very "heavy" by the user. Because of that, the weightiness of the utterance is classified as 1 instead of 0.

- **Sociopragmatic politeness (W)**: Overpolite (+1). Taking into account both the horizontal and vertical distances between the chatbot and the user, as well as the weightiness of the transaction, we believe that the chatbot was overpolite to the user. The excessive use of indirectness and hedging is a piece of evidence that confirms that classification. Even considering that the chatbot may be admitting to record the user's actions (it claims it is offering news according to the user's last reads) the chatbot uses expressions that are too polite for the occasion, as it can be verified by its high pragmalinguistic politeness score (3).

5.2.2.2 Example 2: TechCrunch's First Message

On the second and third examples, we explore the first messages of the TechCrunch and the Wall Street Journal chatbots. These examples illustrate two similar yet distinct approaches to presenting the chatbot and offering the news feed to the user: TechCrunch is more direct and uses less hedging, while the Wall Street Journal is more indirect and more pragmalinguistically polite.

The second example is the first message sent by the TechCrunch chatbot to its user. Figure 5.2 illustrates that message: after the user sends a “Get Started” message (default command for starting a conversation with chatbots on Facebook Messenger), the TechCrunch chatbot replies with the following sentence: “Hi Francisco [user’s name], I’m an AI-based assistant for TechCrunch. I’ll send you a digest of trending stories once a day.”. After that, the chatbot offers the user two options in a *card*: “Ok, what else?” and “No, thanks”.

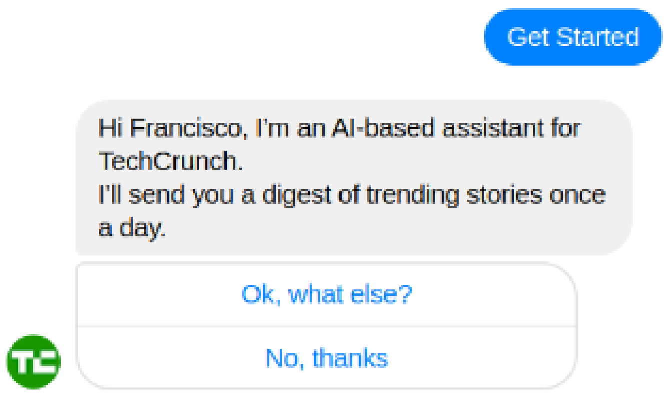

Figure 5.2: Example of utterance from the TechCrunch chatbot. Source: Authors

- **Strategy**: **S1** (*Showing the main feature on the first message*), **S2** (*Guiding the user through a short tutorial during first messages*), **S3** (*Suggesting the next possible set of actions to the user*). As the chatbot offers the stories digest on the first message, which is part of a tutorial, and there are suggestions of answers for the user (“Ok, what else?” and “No, thanks”).

- **Sign Type**: Metalinguistic. This message is an example of a metalinguistic sign, as it explains what the chatbot is (“an AI-based assistant”) and what it can do (“send you [the user] a digest of trending stories once a day”).

- **Speech Act**: Commissive. The chatbot commits itself to “send you [the user] a digest of trending stories once a day”, making a promise to send that digest to the user.

- **Illocutionary function**: Convivial. In this case, the illocutionary goal (send news digest to the user) coincides with the social goal (the chatbot keeps a good relation by offering something to the user).

- **Cooperative Principle**

- **Quality**: Respected. The way the chatbot presents itself and what it can do is truthful.
- **Quantity**: Respected. The chatbot presents two pieces of information (what it is and what it can do). It is not an excessive quantity of information for a first message.
- **Relation**: Respected. As a conversation starter, presenting itself is a good strategy.
- **Manner**: Respected. The message sent by the chatbot is presented in a well-ordered manner.

• **Politeness Principle**

- **Generosity**: Respected. It is offering something to the user (in this case, "a digest of trending stories once a day"), valuing what the user (may) want.
- **Tact**: Flouted. Although it is disguised as an offer, actually it's a request made by the chatbot. It is requesting the user to subscribe to the stories digest in a way that looks like its offering something to the user, as such, the chatbot makes no use of hedges while making its request, and it's not giving a low value to what it wants (i.e. the user to subscribe to the digest). Even so, it gives the user a chance of declining the offer through the *quick reply* "No, thanks".
- We did not find any evidence of Maxims of **Approbation**, **Modesty**, **Obligation of S to O**, **Obligation of O to S**, **Agreement**, **Opinion reticence**, **Sympathy**, nor **Feeling reticence** in this message.

• **Pragmalinguistic politeness**: A little polite (1 out of 5). The utterance is not very polite on the pragmalinguistic scale. It does not make use of hedging nor indirectness. Instead of that, it is very direct in informing the user about what the chatbot is ("an AI-based assistant for TechCrunch") and what it will do ("I'll send you a digest of trending stories once a day"). It is important to note that Leech (2014) states that for an offer (as in the case of the analyzed utterance), which evokes the Generosity maxim, it is more pragmalinguistically polite to be direct, constraining the addressee toward acceptance, giving them less chance to refuse what is being offered. That way, the speaker would be giving more value to O's wants, as stated by the Generosity maxim. But that may not be the case in that sentence: although the chatbot seems to be offering the stories digest to the user, it may also be the case of a disguised request: the chatbot is requesting the user to subscribe to the digest. In that case, the lack of hedging indicates that the utterance is not very pragmalinguistically polite.

- **Vertical distance (P)**: Clerk to a customer (1 out of 3). As we explained before, values for vertical and horizontal distance are fixed for all messages from the same chatbot. After analyzing all of the messages sent by the TechCrunch chatbot during the inspections, we noticed that, overall, the chatbot talks to the user as a salesperson that wants to help a customer, always offering to help the user. Because of that, we classified the vertical distance value as 1.

- **Horizontal distance (D)**: Familiar (1 out of 3). As in the case of vertical distance, the horizontal distance is also fixed for all of the messages from the chatbot. The TechCrunch chatbot sends messages like someone familiar to the user. That is evidenced by the low use of hedging and by the directness of the utterances from the chatbot. Although the chatbot does not make use of slang language, it is overall very informal towards the user (for example, when it cannot understand the user input, it replies a sarcastic remark: "Computers are getting smarter all the time. Just look at me."). Because of that, we classified its horizontal distance as 1.

- **Cost-benefit (R)**: 2 (out of 5). In this utterance, the chatbot is presenting itself ("I'm an AI-based assistant for TechCrunch) and it is offering something to the user ("I'll send you a digest of trending stories once a day"). In a usual person-to-person exchange, an offer would indicate that the speaker is taking on the weightiness of the transaction, and the cost-benefit would be good for the hearer (if I am offering a cup of tea to someone, I am expected to prepare the tea myself). But in the case of a conversation with a chatbot, the cost for the chatbot to do something is almost irrelevant, as it is an automated function. That way, when the chatbot offers to send daily messages with stories, the cost for the chatbot is insignificant, while the cost for the user is greater, as the user will have to abide to a (possible) message spamming from the chatbot. Because of that, we classified the cost-benefit as 2, due to the cost of the user being (possibly) nagged by the chatbot messages.

- **Sociopragmatic politeness (W)**: Underpolite (-1). Considering vertical and horizontal distances and the cost-benefit of the message, we classified the utterance as underpolite. Due to the great cost-benefit of what is being asked (sending daily digests to the user, and possibly spamming them), the chatbot should have been more polite to the user. Although the sentence was very pragmalinguistically polite for an offer (Generosity maxim) by pushing the user to accept what is being offered, when analyzing the sociopragmatic scale, the cost is bigger for the user, as they will have to receive the messages. That way, we classified it as -1 (underpolite) in the sociopragmatic politeness scale.

5.2.2.3 Example 3: Wall Street Journal's First Message

The third and final example is the initial message from the Wall Street Journal chatbot. Figure 5.3 shows that utterance: after the user sends a "Get Started" (which is the default message for starting a conversation with a chatbot on Facebook Messenger), the chatbot first welcomes the user ("Hello! Welcome to WSJ Messenger.") and declares some of its features ("We're here to send you breaking news and live markets data from our award-winning newsroom."). It is interesting to note the use of first-person plural ("we") instead of singular. That is evidence that the chatbot does not have a *persona*, instead, it is an outlet for WSJ's "award-winning newsroom".

Unlike the TechCrunch chatbot (Figure 5.2 on page 87), WSJ does not call users by their names, which can be considered as evidence that the horizontal distance between the WSJ chatbot and the user is greater than the horizontal distance between the TechCrunch chatbot and the user.

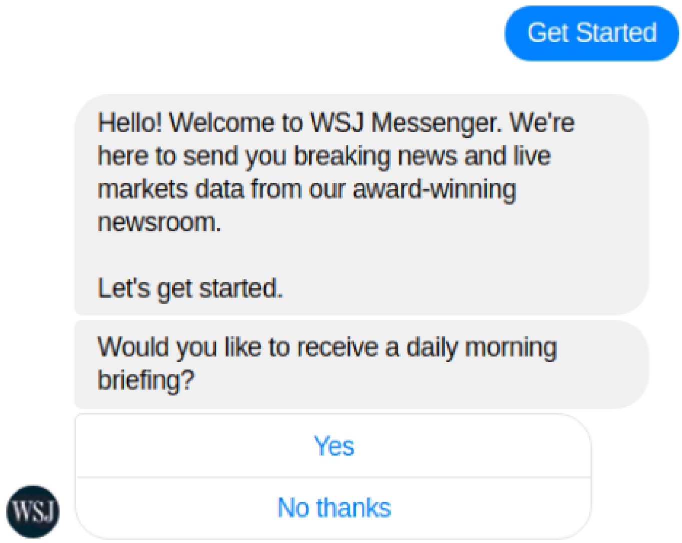

Figure 5.3: Example of utterance from the Wall Street Journal chatbot. Source: Authors

After that, the WSJ chatbot invites the user to "get started", and offers them a daily briefing ("Would you like to receive a daily morning briefing?"), offering, in a *card*, two options for answer: "Yes" and "No thanks". That contrasts to TechCrunch's message offering its daily digest ("I'll send you a digest of trending stories once a day."): instead of just informing the user that it will send the stories in a very direct way, WSJ takes a more indirect approach, asking the user whether they "would like to receive" the daily briefing. These different approaches for offering basically the same feature and presenting themselves illustrate how distinct chatbots can be designed with different familiarity (horizontal distance) levels in mind.

- **Strategy**: **S1** (*Showing the main feature on the first message*), **S2** (*Guiding the user through a short tutorial during first messages*), and **S3** (*Suggesting the next possible set of actions to the user*). The chatbot is showing its main feature (the daily morning briefing) on the first message. This message, along with the next two sent by the chatbot can be considered a short tutorial of some of the features it offers. Finally, the chatbot also suggests to the user two answers to its inquiry: "Yes" and "No thanks".

- **Sign Type**: Metalinguistic (first sentence) and static (second sentence). The first part of the message explains to the user what is the general purpose of the chatbot ("We're here to send you breaking news and live markets data from our award-winning newsroom."), characterizing a metalinguistic sign. But, on the second part, the chatbot offers one of its features ("Would you like to receive a daily morning briefing?"), and that is a case of a static sign.

- **Speech Act**: Assertive (first sentence) and commissive (second sentence). In the first sentence, the chatbot is presenting itself, so it is the case of an assertive speech act. Then, the chatbot offers something (breaking news, live market data, and a morning briefing) to the user. That way it is committing itself to something, and the sentence can be classified as a case of commissive speech act.

- **Illocutionary function**: Convivial. The chatbot's illocutionary goal (sending market data, breaking news, and morning briefing) coincides with its social goal (maintaining a good relationship with the user, so they keep talking to the chatbot and reading WSJ stories).

- **Cooperative Principle**

  - **Quality**: Respected. All of the information presented by the chatbot are true. The Wall Street Journal has indeed received many awards and the chatbot will send the user news stories and market data.

  - **Quantity**: Respected. The chatbot presents an adequate quantity of information in this utterance, informing a few things it can do.

  - **Relation**: Respected. The information sent by the chatbot presents the chatbot itself and some of its functions, which is a good way of presenting oneself at the beginning of a conversation.

  - **Manner**: Respected. The chatbot message is presented in a well-ordered manner.

- **Politeness Principle**

  - **Generosity**: Respected. The chatbot is actively offering a morning briefing to the user. Besides that, it also implies that it can send the user breaking news and live markets data.
  - **Tact**: Respected. The use of indirectness ("would you") is evidence that the chatbot is respecting the Tact maxim, avoiding imposing itself on the user, as it focuses on what the user wants.
  - **Modesty**: Violated. The chatbot is not very modest by claiming to have a "award-winning newsroom".
  - We did not find any evidence of Maxims of **Approbation**, **Obligation of S to O**, **Obligation of O to S**, **Agreement**, **Opinion reticence**, **Sympathy**, nor **Feeling reticence** in this message.

- **Pragmalinguistic politeness**: Polite (2 out of 5). The chatbot makes use of indirectness when offering the morning briefing to the user. That is indicative of its pragmalinguistic politeness level. It is not as polite as the message from Example 1 (Figure 5.1, on page 84), but it is more polite than the message from Example 2 (Figure 5.2, on page 87). That way, we classified it as 2 on the pragmalinguistic politeness scale.

- **Vertical distance (P)**: Clerk to a customer (1 out of 3). The vertical distance, which is fixed for the chatbot as a whole instead of just this message, is decided after analyzing all of the utterances from the chatbot. In the case of the WSJ chatbot, we noticed it is similar to an assistant for an executive, always trying to help. There is a distance between the chatbot and the user, but it is not very big. That way, we classified the vertical distance as 1.

- **Horizontal distance (D)**: Acquaintance (2 out of 3). As said before, the horizontal distance is classified in a similar way to the vertical distance, that is, after analyzing all messages from the chatbot. The WSJ chatbot makes great use of indirectness and, sometimes, hedging when communicating to the user. Out of the three chatbots analyzed in the first part of this work (CNN, TechCrunch, and WSJ), it is the most (horizontally) distant from the user. Because of that, we classified it with a horizontal distance of 2.

- **Cost-benefit (R)**: 2 (out of 5). This message is similar to the message from the second example (Figure 5.2 on page 87), as both are offering a daily digest of news to the user. That way, we classified them with the same value for cost-benefit: 2. That is because, differently from what happens in person-to-person communication, when dealing with computers, the weight of what the computer will have to do in the exchange is almost nullified; because of that, the weight is passed on to the other part, i.e., the person/user. In the case of a daily digest, the user could be overwhelmed by undesired spamming from the chatbot.

- **Sociopragmatic politeness (W)**: Politeness adequate for the situation (0). After analyzing the vertical and horizontal distances adopted by the chatbot and the cost-benefit for this message, we noticed that, while the cost-benefit is somewhat high, the chatbot makes use of indirectness while offering the daily summary. The use of indirectness gives the user chance to politely deny what is being offered. Although the chatbot would not be offended by a denial from the user, as it is mimicking the way humans communicate, it is interesting to see the use of indirectness. Because of that, we concluded that the level of politeness used by the chatbot is consistent with its vertical and horizontal distances, hence, we classified it as 0 (adequate politeness) in the sociopragmatic politeness scale.

5.2.3 Insights from the Analysis

After analyzing and classifying messages from the three chatbots, we were able to reach some insights. These insights are described in this subsection.

5.2.3.1 Cooperative Principle

We noticed that in the majority of cases when the Relation maxim (from Gryce's Cooperative Principle) was violated, it happened at the same time as a communication breakdown. Figure 5.4 on the following page exemplifies this: the user is trying to set up the breaking news delivery. When the user types "breaking news", expecting to read the breaking news, the TechCrunch chatbot instead replies with a sarcastic remark "Computers are getting smarter all the time. Just look at me."; later, the user tries again, asking to "setup breaking news", instead of helping them set up the delivery of breaking news, the chatbot tries to look for news stories about "setup breaking news"; finally, the user tries one more time to "Setup Breaking News", only to be answered that the chatbot is "Living the dream".

These examples depict cases in which communication breakdowns happened, that is, the user was expecting something, based on previous interactions with the chatbot (in these cases, the user knows that the chatbot has a breaking news feature, as well

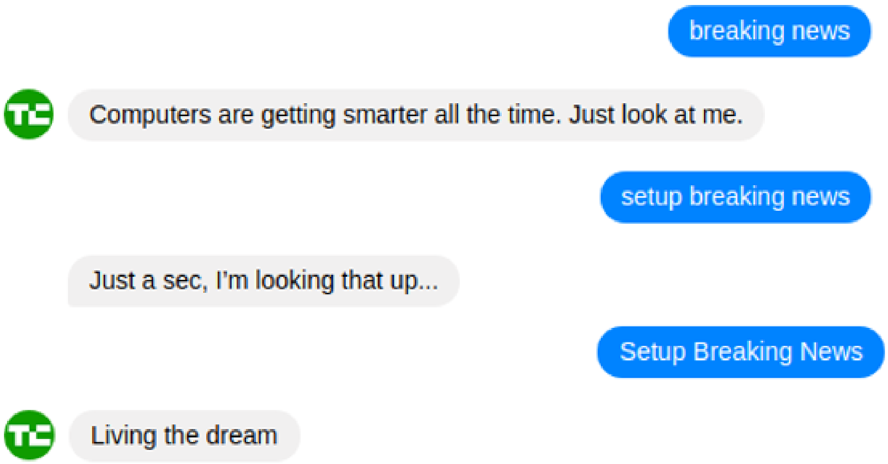

Figure 5.4: Example of the TechCrunch chatbot violating the Relation maxim. Source: Authors

as another feature for setting up their delivery, as shown on Figure 5.5, displaying the answer the user was expecting), but is surprised with a different reaction from it. These examples are also cases of the chatbot violating the Relation maxim, as the chatbot‘s reply does not make a relevant contribution to the conversation, considering the user's previous utterance. There is no apparent reason for the chatbot to be flouting the maxim, so we can discard the hypothesis of a conversational implicature taking place here.

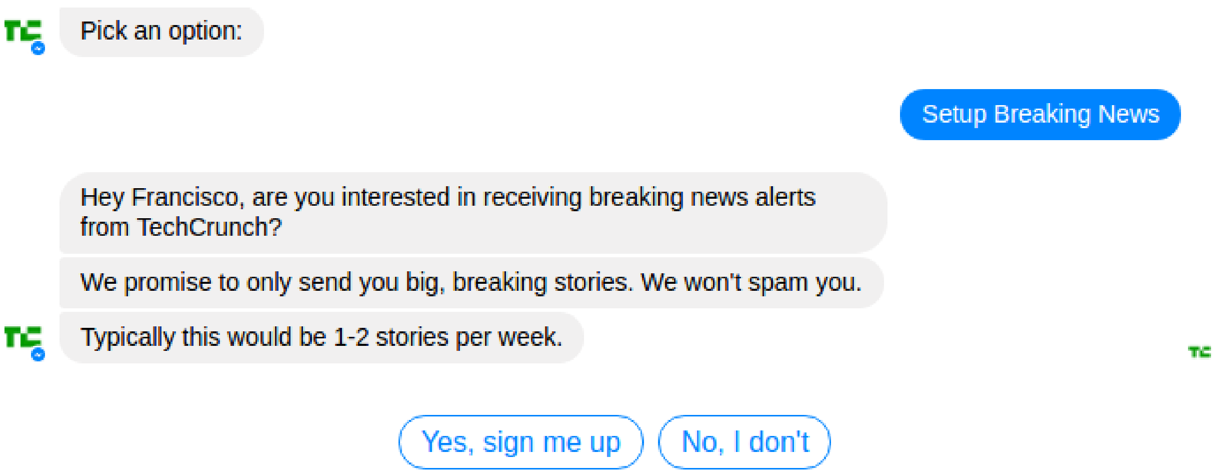

Figure 5.5: User setting up the breaking news delivery on TechCrunch chatbot. Source: Authors

As we said before, most times that the Relation maxim is not observed happen when the chatbot cannot understand what the user is saying. On the three analyzed chatbots, when that happens they usually try to search for some news stories containing the user input in order to mend the communication breakdown that happened and to avoid violating the Relation maxim. That way, in the analyzed chatbots, we may consider that most cases of Relation maxim violation happened when the chatbot designer failed to consider the specific interaction path the user took.

It is possible that there are cases in which the Relation maxim is deliberately flouted. The chatbot designer may opt to flout the maxim in order to surprise users with a response they are not expecting, but that was not observed during the analysis.

An interesting case regarding the Relation maxim happened in the WSJ chatbot, in which the chatbot asks for clarification in order not to violate the Relation maxim and avoid a possible communication breakdown. Figure 5.6 on the following page shows the case: the user wants to know the key financial metrics for Apple, so they ask “APPLE key metrics”; the chatbot checks that there are two companies with Apple in their names: Apple Inc (AAPL) and Apple Hospitality REIT Inc (APLE), so, in order to confirm which Apple the user wants to know about, the chatbot shows a list of these companies, that way, the chatbot avoids giving information about the wrong company and violating the Relation maxim in the process; in the sequence, the user selects the correct company (Apple Inc); and the chatbot replies with the requested metrics. In this case, it is also interesting to note that the user’s first message violated the Manner maxim, as it was ambiguous regarding the company name (even if unintentionally – i.e. user might not have known there was more than one company with Apple in its name).

Regarding Manner and Quantity maxims, few violations were noticed during our analysis, those that happened were mostly related to the chatbot failing to inform the user about different ways to access a feature, or exhibiting a very large list of quick-replies. It seems that cases of violation of these two maxims are mainly related to bad design choices made by the chatbot designer, such as a badly written or confusing sentence (violating Manner), or sending too many messages at once, or presenting too many or too few options for the user at once (violating Quantity).

The Quality maxim was not respected only once during our analysis: when the user asked “Are you a real person?”, the TechCrunch chatbot replied “Of course! Do you still have any questions?”, as shown in Figure 5.7 on the next page. That was clearly a joke, but it raises the question: should we consider that the chatbot can believe something is true or not (as the Quality maxim states: “Do not say what you believe to be false”)? I do not believe we should consider the chatbot to have its own set of beliefs, as the chatbot will always do what it was designed to do. But the chatbot’s designer may say (through the chatbot) something they believe to be false (as in the example on Figure 5.7 on the following page).

Considering a hypothetical case of a chatbot that makes use of an external source for looking up results, suppose there is incorrect information in that external source, would that be a violation of the Quality maxim? The chatbot designer was probably expecting the information being relayed by the chatbot to be true. It is really difficult to

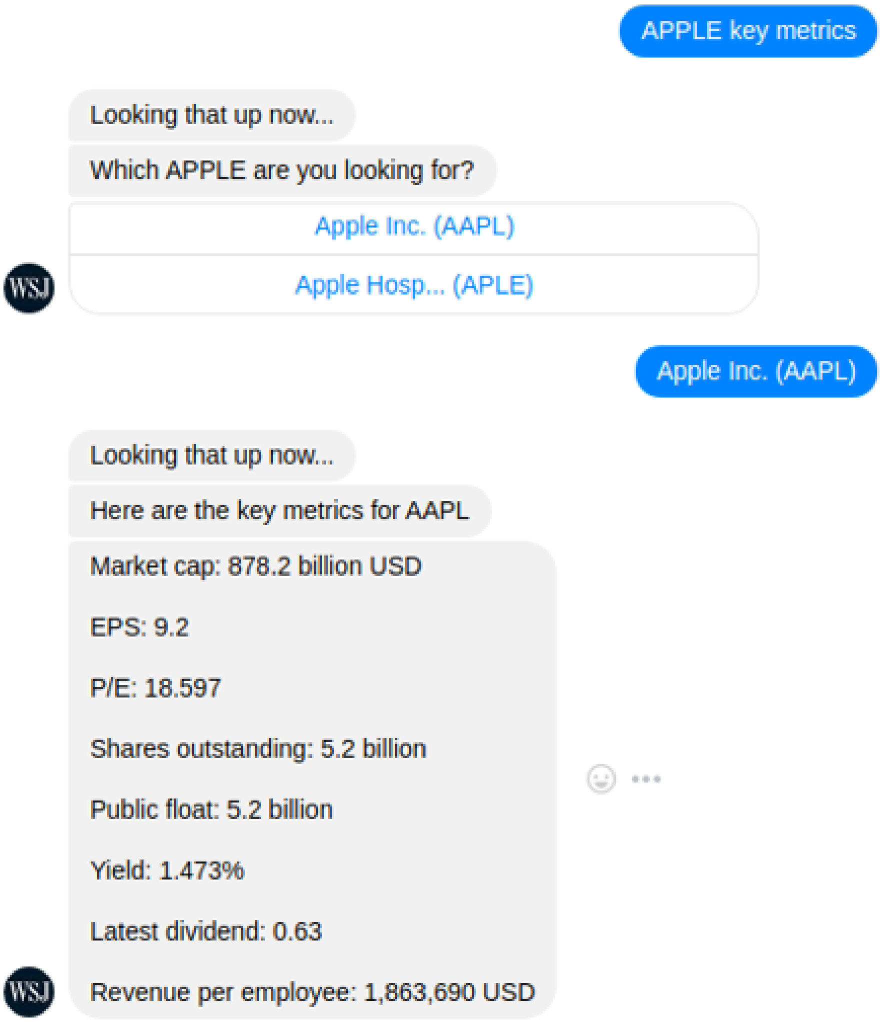

Figure 5.6: WSJ chatbot asking user to clarify a query. Source: Authors

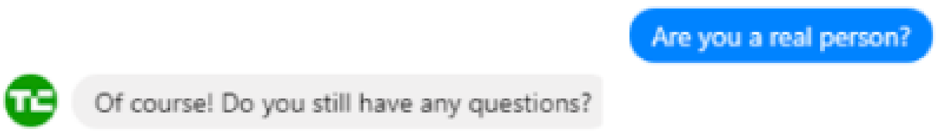

Figure 5.7: TechCrunch chatbot says it is a real person. Source: Authors

analyze that, as we would not be able to check whether the mistake was intentional or not. Nonetheless, the effect on the user during the conversation is the same: the chatbot is not following the Cooperative Principle, as it is violating the Quality maxim (in case of false information) or the Relation maxim (in case of an authentic, but non-related, information being relayed).

That way the Cooperative Principle's maxims may help us understand some factors that may be behind communication breakdowns during chatbot interaction: when the breakdown occurs, a violation of some of the maxims will also take place. During the inspections of these three chatbots, breakdowns associated with violations of Manner, Quantity, and Quality were mostly derived from badly written chatbot messages or bad conversational design (such as sending too many messages or options at once or failing to inform alternative ways to access a feature). These kinds of breakdowns are mostly related to the chatbot design itself and the decisions their designers took during their creation. Thus, if chatbot designers take the maxims into consideration, they may able to avoid some of the breakdowns we found during our analysis.

On the other hand, breakdowns related to violations of the Relation maxim may be derived to questions other than the conversational design, such as the Natural Language Processing failing to correctly interpret what the user is saying, leading to a non-sequitur in the conversation (such as on Figure 5.4 on page 94, for example) and communications breakdown.

5.2.3.2 Politeness Principle

Regarding Leech's Politeness Principle maxims, we noticed that the most respected one is the **Generosity** maxim. That happens because the analyzed chatbots are constantly offering something to their users. They may do that openly, in the messages they sent, by telling the user about a feature, or by showing *quick replies* with links to their features. This last case may be interpreted as the chatbot saying "Here are all those things I can do for you". Because of that, the Generosity maxim appears to be very related to the strategy **S3** (*Suggesting the next possible set of actions to the user*) from Chapter 4 - "Communicative Strategies Investigation".

All three analyzed chatbots used the Generosity maxim in their first messages, as they offered some of their features to users. That makes **S1** (*Showing the main feature on the first message*) strategy closely related to the Generosity maxim. The same can be said about **S2** (*Guiding the user through a short tutorial during first messages*) strategy, as during the tutorial the chatbot shows the user how to use its features, constantly offering these features to the user.

We have found evidence of the **Tact** maxim in several sentences throughout the analyzed chatbots. Most times the chatbot offered something to the user (not through *quick replies*) could be considered as a request in disguise (i.e. a directive speech act), which is made clear by the use of hedges and neg-politeness when making the supposed offer. That happens because part of the interest of the chatbot designer is that the user keeps talking to it, and an easy way to prolong that conversation is by making the user sign up to a news feed, so the chatbot will keep sending daily messages, lowering the chances of being forgotten by the user. The first message from the WSJ chatbot (Figure 5.3 on page 90), and the set up of the breaking news on the TechCrunch chatbot (Figure 5.5 on page 94) are examples of that: the chatbots offer something – "Would you like to receive a daily morning briefing" in the WSJ, and "... are you interested in receiving breaking news alerts from TechCrunch?" in TC. Both the offers are also requests, making use of neg-politeness ("would you like"), TechCrunch chatbot goes beyond that and adds "We promise to only send you big, breaking stories" and "We won't spam you.", evidencing that they also took into consideration the cost-benefit for the user. They both give users the chance to politely decline the request, and TC indicates in its utterances the intent to minimize the cost and maximize the benefits to users (e.g. promising it will only send relevant information).

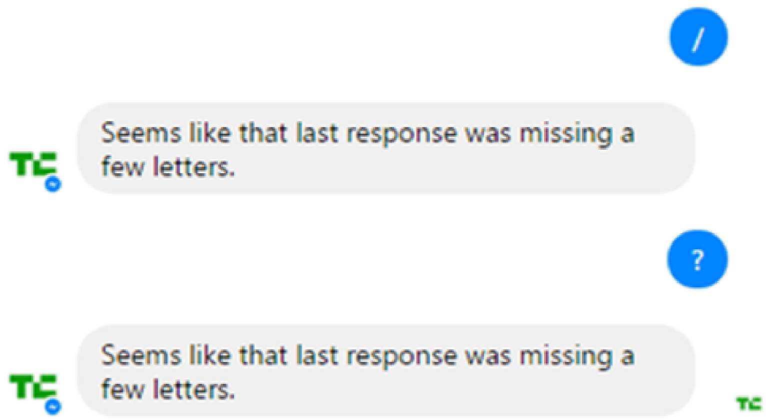

Figure 5.8: TechCrunch chatbot requesting the user to try again. Source: Authors

Tact maxim was also noticed in a message the chatbot sent when it did not understand the user's utterance. When the user sends a message containing only a question mark or a slash, the TechCrunch chatbot will ask the user to try again, as seen in Figure 5.8. The way the chatbot does it is indicative of the Tact maxim, as the chatbot is rather indirect ("Seems like...") when informing the user that the input was not valid and the chatbot could not understand it. The chatbot is implicitly asking the user to try again with other words.

TechCrunch also respected the Tact maxim when asking for feedback. As Figure 5.9 on the next page shows, TechCrunch says "We're happy to hear any feedback", when requesting the user to type their feedback to the chatbot. In Figure 5.10 on the following page, TechCrunch is also making use of the Tact maxim by saying it would "love to hear your feedback", instead of requesting it directly. As providing feedback is more costly to the user, that will have to formulate and type it, it makes sense that the chatbot avoids requesting it bluntly.

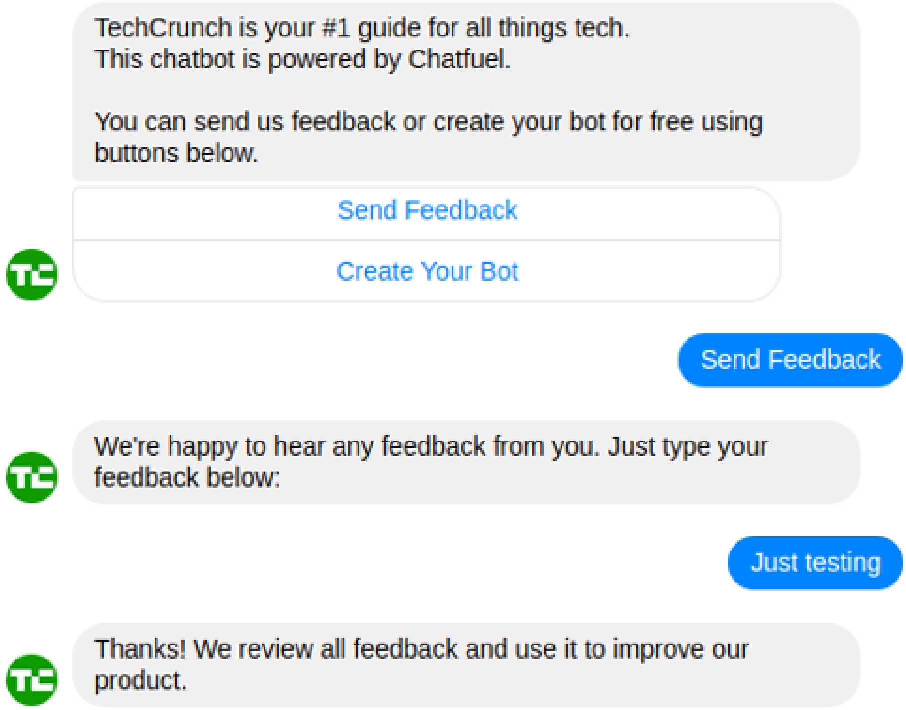

Figure 5.9: TechCrunch feedback options. Source: Authors

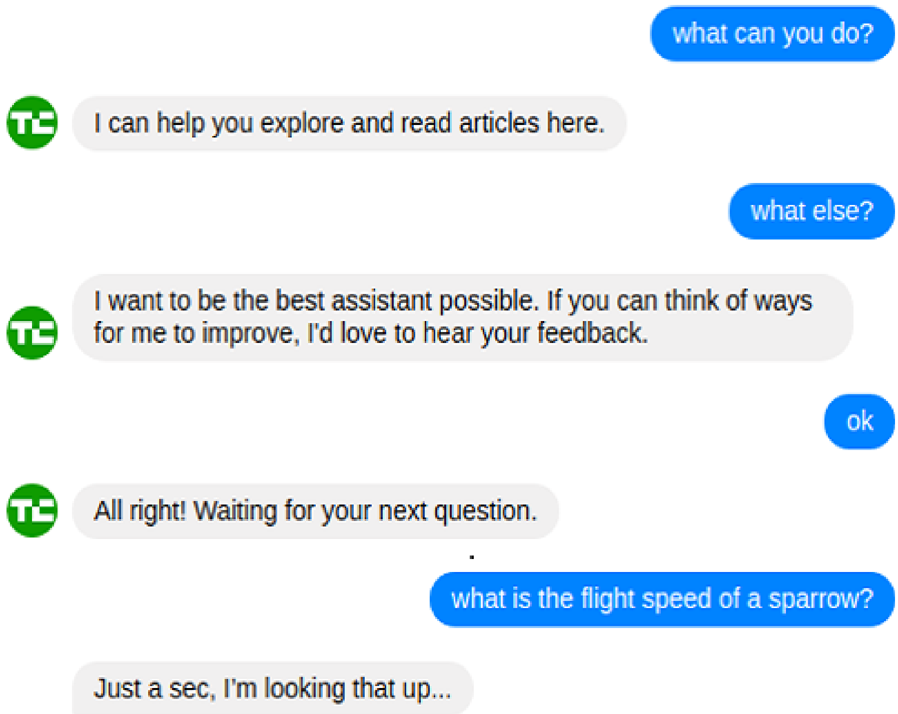

Figure 5.10: TechCrunch chatbot keeping a good conversation and asking for feedback. Source: Authors

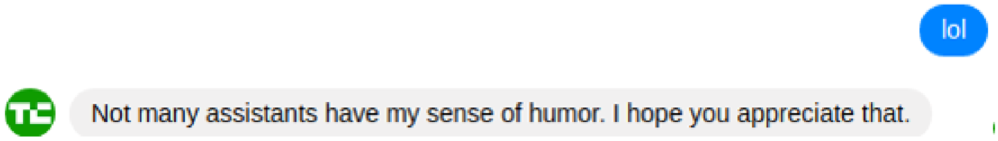

Figure 5.11: TechCrunch not being very modest. Source: Authors

The **Approbation** maxim was noticed only in the small talk messages from the TechCrunch chatbot. These small talk messages are not related to any feature from the chatbot and seem to be there mostly for making the chatbot feel more amicable and like a real person instead of just a feed of news stories. Figure 5.16 on page 106 shows an example of this maxim: the chatbot tells the user that "You have a *great* laugh.", making use of pos-politeness and enhancing the pragmalinguistic politeness of compliment.

The **Modesty** maxim was noticed on a few occasions. When the TechCrunch chatbot asked for feedback (Figure 5.10 on the preceding page), it put a low value to its qualities by saying it "want to be the best assistant possible", implying it still can be improved.

Nonetheless, in some cases, the Modesty maxim was violated, as in the case of the first message from WSJ chatbot, analyzed in the previous section (5.2.2.3 - "Example 3: Wall Street Journal's First Message", on page 90), in which the chatbot talks about its "award-winning newsroom", although we may consider that the chatbot is just asserting that its newsroom was awarded and respecting the Quality maxim. In another case (Figure 5.11), when answering some small talk, the TechCrunch chatbot says "not many assistants have my sense of humor" which is not very modest.

The PP maxim of **Obligation of S to O** was also found to be related to some strategies found in Chapter 4 - "Communicative Strategies Investigation". As the Obligation of S to O maxim is mainly about apologies and thanks, it appeared in our analysis mainly when the chatbot apologized for failing to understand what the user said. That way, the Obligation of S to O maxim can be related to strategies **S9** (*Showing the main menu or the most frequent features when user asks for help*) and **S3** (*Suggesting the next possible set of actions to the user*), as the chatbots usually say they are sorry along with showing their features when they cannot understand the user requests. Figure 5.12 on the following page shows the WSJ chatbot apologizing after not being able to understand the user's command and showing a menu with options after that.

The Obligation of S to O maxim was also found on a few occasions when the chatbot thanked the user for using the chatbot or for leaving some feedback, as shown in Figure 5.9 on the previous page, in which the TechCrunch chatbot thanks the user for giving feedback about the interaction. The TechCrunch chatbot also makes use of the Obligation of S to O maxim when replying to some small talk: as Figure 5.16 on page 106

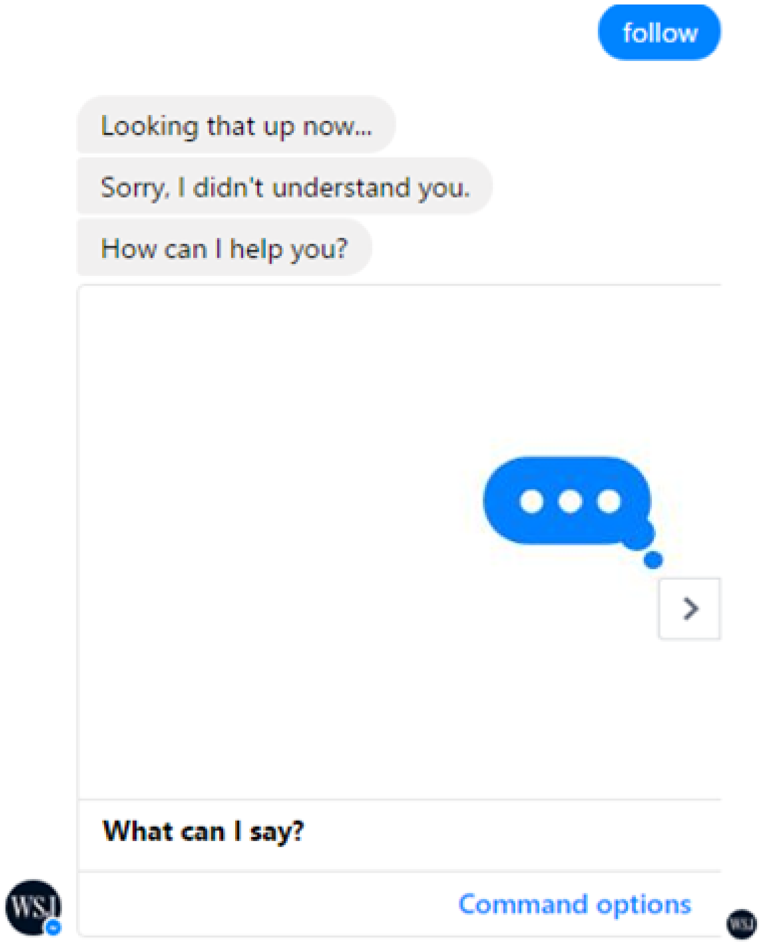

Figure 5.12: Wall Street Journal chatbot apologizing to user. Source: Authors

shows, when asked "how are you?", the chatbot thanks the user by replying "Wonderful as always. Thanks for asking.".

Regarding the **Obligation of O to S** maxim, that is, responses to thanks and apologies, we only noticed it on three occasions. The TechCrunch chatbot does it when replying to "thanks" and some variations of it; and also when the user asks for help, that is responded as the same way as replying to a "thank you" message, Figure 5.13 on the next page shows these examples. The last case is from the CNN chatbot: when the user sends a thumbs up to the chatbot, it appears to consider it as a compliment and replies "Anytime. That's what I'm here for." (Figure 5.14 on the following page).

During the analysis of the three chatbots, we found no signs of the last maxims of the Politeness Principle: Agreement, Opinion reticence, Sympathy, and Feeling reticence maxims.

The **Agreement maxim** (Give a high value to O's opinions) was not noticed probably because none of the chatbots ask for the user's opinions, nor try to agree (or disagree) with them. Maybe the chatbot designers could take that maxim into account

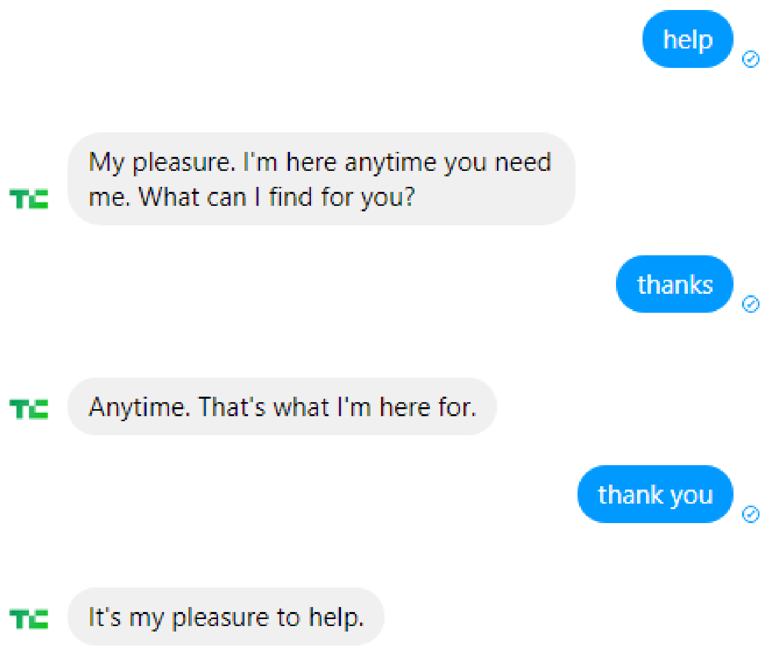

Figure 5.13: TechCrunch responding to thanks. Source: Authors

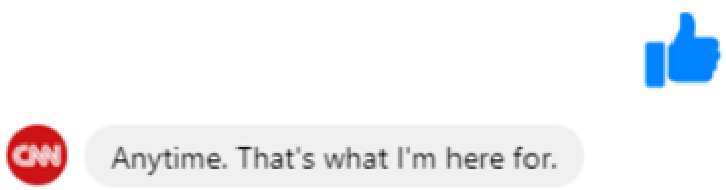

Figure 5.14: CNN chatbot replying to a thumbs up. Source: Authors

when replying to feedback from the user. Out of the three analyzed chatbots, only the TechCrunch chatbot has a feedback option. But, as Figure 5.9 on page 99 shows, after receiving the user's feedback, TechCrunch only thanks the user for it, making no comment about the feedback itself. The counterpart of the Agreement maxim, the **Opinion reticence** maxim (Give a high value to O's opinions), is related to avoiding to give one's views in an opinionated manner (Leech, 2014, p. 204). As the analyzed chatbots never express opinions whatsoever, the Opinion reticence maxim was also not noticed during the analysis.

The **Sympathy** maxim was not noticed during the analysis, however, that could be associated with the domain of the chatbots (news). One could easily think of other domains in which a chatbot could make use of the Sympathy maxim, for instance in a chatbot related to some kind of emotional support, or even one that, in whatever domain, involved a quiz function. In this case, the chatbot could make use of the Sympathy maxim by congratulating users for guessing the right answer, or encouraging them when they got some wrong answers.

Finally, the **Feeling reticence** maxim was also not found in the analyzed chatbots. Once again, that could be related to the news domain, which would not be expected to have much communication about feelings involved. However, the Feeling reticence maxim is associated with the speaker suppressing their feelings. Thus, it is valid to consider whether chatbots can have any feelings at all to be suppressed. At any rate, if we consider that the designer could program the chatbot to consider Sympathy, there might be situations in which the chatbot that does talk about feelings would consider not bringing it up, and thus be considered to display the Feeling reticence maxim. Maybe a chatbot with a very strong personality, written akin to a character in a play could also display the Feeling reticence maxim.

5.2.3.3 Pragmalinguistic and Sociopragmatic politeness scales

The three analyzed chatbots were very constant in the type of language they used throughout the conversation. The WSJ chatbot was the most formal of the three, and, in general, TechCrunch was a little bit more informal than the CNN chatbot. Regarding the **pragmalinguistic scale** of politeness, the TechCrunch messages were mostly classified as a little polite (1) in that scale, with few outliers ranking higher when the chatbot was offering its daily digest and when replying to a "help" message from the user. The CNN chatbot, by its turn, had some of its utterances classified as the same value of the TechCrunch, while a few other times was more pragmalinguistically polite, especially when suggesting a news story to its user (see the example of Figure 5.1 on page 84). The Wall Street Journal chatbot was the most pragmalinguistically polite during our analysis, having more messages with higher marks in the pragmalinguistic politeness scale, particularly during interactions asking the user for confirmation or to take an action.

In order to classify the chatbots utterances in the **sociopragmatic scale**, we first classified the chatbots regarding their vertical (power relation) and horizontal (familiarity) distances relating to the user. As we said, that distance was estimated after looking at all of the chatbots' messages and inferring their distances from the way they address the users throughout the interaction.

As explained in section 5.2.2 - "Classification Examples" on page 82, the classification of the chatbots on these aspects was the following: TechCrunch received the classification as a salesperson helping a customer for vertical distance, and as a person familiar to the user for horizontal distance. The CNN chatbot was also classified as a salesperson to customer relation for vertical distance, as it seemed to be not as in the same level of the user. For the horizontal distance, it was classified as an acquaintance, as it does not behave as very familiar to the user. Finally, the WSJ chatbot was also classified with a vertical distance of a salesperson helping a customer, and the same classification as the CNN chatbot for horizontal distance, that is, of an acquaintance to the user.

Regarding the cost-benefit involved in the utterances of the chatbots, in most cases, there was no cost involved whatsoever, as most of the messages did not ask the user to do anything, particularly the small talk messages from TechCrunch or when the chatbot was delivering something the user had asked. The cases with some cost involved were those in which the chatbot was asking the user to commit to something, such as subscribing to a daily digest. This case, as previously discussed in section 5.2.2 - “Classification Examples”: although the chatbot is offering to do something to the user, in this case, send a daily briefing of news stories, the cost weights more on the user than on the chatbot, as the user can become bothered by those messages, and there is virtually no cost for the chatbot, as it is not really a person rather a piece of software.

Concerning the sociopragmatic scale of politeness, the WSJ chatbot had very consistent results: all of its messages were classified as having politeness appropriate for the situation, meaning it maintained a similar level of politeness throughout the conversation, and when the cost-benefit involved in the sentence was higher, it was more polite than usual.

The TechCrunch chatbot had most of its messages classified as having politeness appropriate to the situation, although there were a few messages classified as overpolite when offering the daily digest – these overpolite messages were also classified as very pragmalinguistically polite. There was also a single message classified as underpolite: the very first message, in which the chatbot informs the user that it will send the daily digest of news stories, as detailed in subection 5.2.2.2 - “Example 2: TechCrunch’s First Message” on page 86.

The case of the CNN chatbot is similar to TechCrunch’s. Most messages had politeness appropriate to the situation, one message was classified as underpolite (an apology message with an emoji, very out of character for an otherwise very serious chatbot, as seen on Figure 4.2 on page 33), and another one was classified as overpolite (the excessive use of hedges on the example in subsection 5.2.2.1 - “Example 1: CNN’s News Story” on page 83).

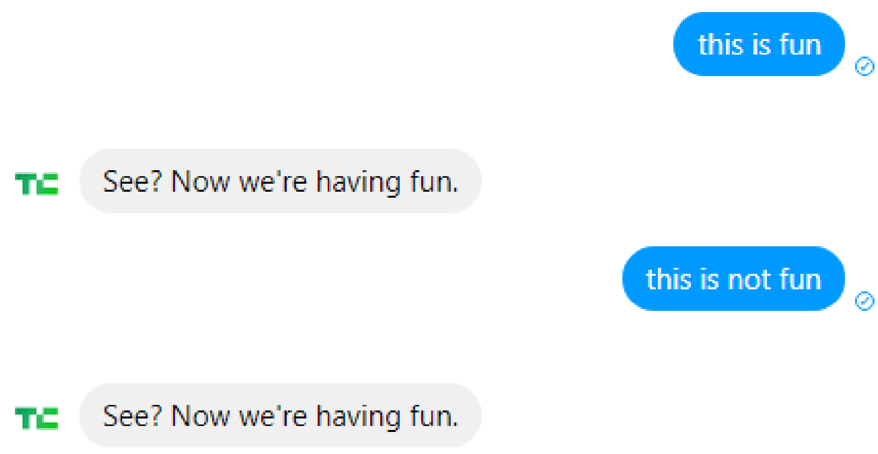

Figure 5.15: TechCrunch chatbot giving the same answers to opposing questions. Source: Authors

5.3 Considerations

An aspect that arose when using these theories to analyze chatbot interaction is that some utterances from the chatbot may make sense in a certain context, but not in another. Figure 5.15 shows that case: the same sentence ("See? Now we're having fun.") was used as a reply to very distinct utterances from the user, in the first case ("this is fun"), it was a valid answer, but in the second case ("this is not fun"), it was nonsense, making it look like the chatbot was mocking the user. One could notice this problem without the use of the maxims, but they are useful for understanding why the problem happens in a communicative way (that is, some of the maxims were violated).

Another example shows that a change in the context of the conversation may turn the same sentence from the chatbot from a valid reply that complies with the CP and PP to a bad one that violates some maxims. The TechCrunch chatbot appears to reply to every input from the user that contains the word "how" with the sentence "Wonderful as always. Thanks for asking.". That approach produces a good interaction in the example of Figure 5.16 on the next page, in which the chatbot is replying to the user saying "How are you?". But in the case of Figure 5.17 on the following page, the results are not very good: the user asks what the chatbot can do, to which it replies "I can help you explore and read articles here.", the user then wants to know *how* the chatbot can help them explore and read articles, so they ask "how?", referring to the previous reply; the chatbot misinterprets that sentence and replies "Wonderful as always. Thanks for asking.", violating the Relation maxim and causing a communication breakdown.

So, which case (if any) should we consider when inspecting the chatbot? Should

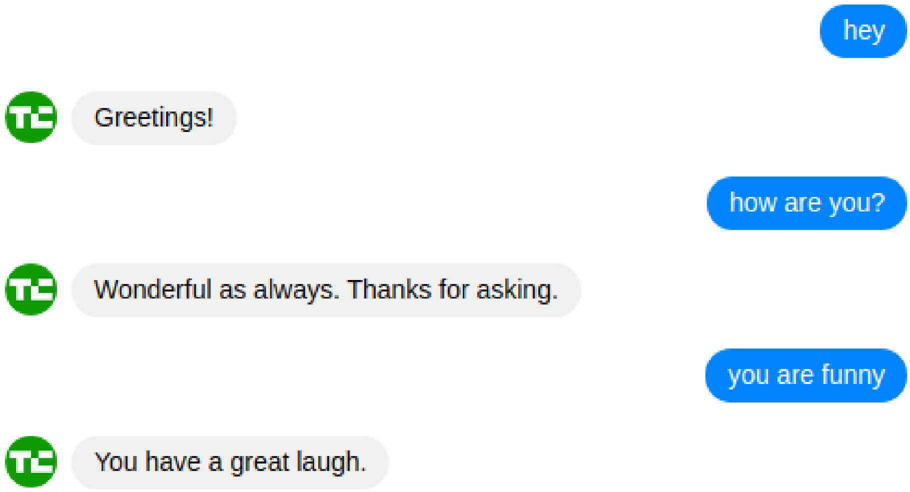

Figure 5.16: Example of TechCrunch chatbot keeping a good conversation. Source: Authors

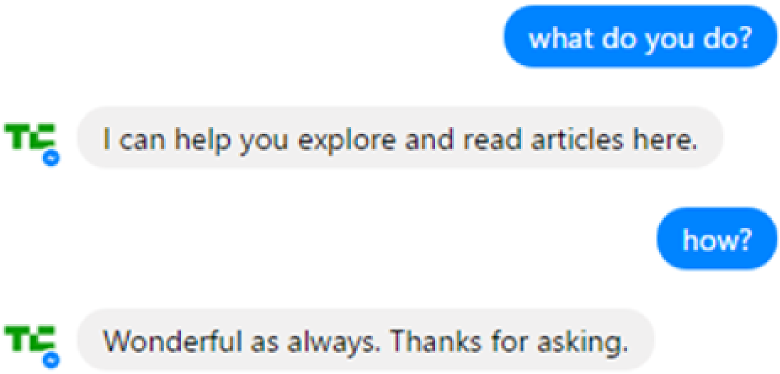

Figure 5.17: Example of TechCrunch chatbot misinterpreting the user. Source: Authors

we consider what the chatbot designer has probably intended or what was shown to the user? To chatbots happen the same as to other types of software: only a segment of their designer's semiosis during the design time is encoded and frozen in the chatbots. While the user's semiosis may change through time, the chatbot's will not. This is an important point that should be investigated in further works.

5.4 Conclusions

In this chapter, we continued our research by using a few concepts from Linguistics in order to complement the strategies for conveying features and the sign classes we discovered while applying the Semiotic Inspection Method to selected chatbots.

We then proceeded to check whether it was viable to use the maxims of the Cooperative Principle and the Politeness Principle to chatbots. In order to do so, we also had to check the chatbot messages for other aspects, such as speech acts, illocutionary function, pragmalinguistic and sociopragmatic politeness, as well as vertical and horizontal distances and cost-benefit.

We had to adopt few premises for using these theories in chatbots, as they were originally intended to analyze person-to-person communication and not person-to-chatbot communication. For in the former both parties have equivalent cognitive and interpretative capacities, while in the latter, the chatbot's semiosis is frozen in time, and its communicative capacities are limited to what was foreseen and programmed by its designer. So this difference in the communication parties implies in some considerations regarding the maxims: (1) what can be considered as having a great cost to a human being can have its cost almost nullified to a chatbot (as its a machine); (2) what can be considered as 'feelings' when talking about a chatbot; and (3) can we consider that the chatbot *believes* what it says?

We also had to adapt the pragmalinguistic and sociopragmatic scales of politeness (including the vertical and horizontal distances and the cost-benefit factors) in order to classify the chatbots' utterances, so they could be compared in these terms. Another premise was regarding how to consider the chatbot's vertical and horizontal distances to the user. We decided to take into consideration the way the chatbot addressed the user during all of the interaction and infer from that how its designer thought how it should treat users based on its language, that is, if it is more polite, more formal, or if it makes use of informal language, for example.

After that, we moved ahead to analyze and classify all of the messages from the first three chatbots we had applied SIM to in Chapter 4 - "Communicative Strategies Investigation". These chatbots were chosen for they are successful and from a similar background, indicating that their designers made good decisions when creating them. Furthermore, as we had inspected them thoroughly before, we already had all of their messages recorded as signs evidence, which facilitated our work.

An important caveat about this work is that the classification of the chatbots' messages was not triangulated, and the only researcher that classified them, although fluent in English, is not a native speaker, so it is possible that some subtleties of the messages may have gone unnoticed during the analysis.

We noticed a possible relation between violations of the Relation maxim from the Cooperative Principle and cases of communication breakdowns. Usually, when these happened a violation of Relation also took place, that is, the chatbot failed to understand what the user was requesting. The approach for the three analyzed chatbots in these cases (when the chatbot could not interpret what the user input was) was to search for news stories containing the user's utterance as a string, in order to mend the breakdown and avoid violating the Relation maxim.

The Relation maxim could also be noticed when the Wall Street Journal chatbot asked users to clarify their input, in order to not violate the maxim.

In general, Relation maxim's violations are associated with interaction paths the chatbot's designer failed to predict, so users end up receiving a message that makes no sense in that context (being it an error message or a message originally intended for another occasion). Violations on Manner and Quantity are more related to bad choices while writing the chatbot's messages, such as sending a confusing message or too many messages at once, respectively. These violations can cause communication breakdowns during the interaction.

Regarding the Quality maxim, there is a question of whether the chatbot could (or not) be considered to believe the things it says. Since the chatbot's semiosis is frozen in time, we may say that the chatbot considers true what its creator is thinking at design time. But if the chatbot is dependent on external resources (for news stories, for example) should we consider that the chatbot designer deems these resources as truthful as well? This can be an indication of multiple discourses on the same system (the chatbot designer's and the resource curator's). These questions should also be further analyzed in future works.

About the Politeness Principle maxims, we noticed that some of them can be related to the strategies from Chapter 4. Mainly the maxim of Generosity being linked to strategies **S1** (*Showing the main feature on the first message*), **S2** (*Guiding the user through a short tutorial during first messages*), and **S3** (*Suggesting the next possible set of actions to the user*). The Tact maxim also being identified in requests disguised as offers and in apologies. Modesty, in a few assertive messages. The maxim of Obligation of S to O is related to apology messages, so we could associate it to **S9** (*Showing the main menu or the most frequent features when user asks for help*) and **S3** (*Suggesting the next possible set of actions to the user*). Approbation and Obligation of O to S maxims were usually noticed on small talk messages. We did not find evidence of Agreement, Opinion reticence, Sympathy nor Feeling reticence maxims during our analysis.

The aspects studied in this Chapter (Speech Acts and the Cooperative and Politeness principles) can be used to better account for communication breakdowns that may occur during the interaction. We have shown how some cases of breakdowns can be related to the maxims. That can be useful for chatbot designers to avoid such breakdowns by taking into account the maxims during design time. Designers can also consider the pragmalinguistic and sociopragmatic politeness scales when writing the chatbot's utterances so they can use adequate politeness to the situation, as well as using a consistent amount of politeness throughout the interaction. With that, we believe users' experience with chatbots can be improved.

Further investigation is necessary, such as triangulating the classification results and expanding the analysis to chatbots of other contexts.

In the next chapter, we explain the user tests we made in order to check users' perception of different ways of interacting with chatbots and some strategies and sign classes from Chapter 4.

Chapter 6

User tests

One of the conclusions we derived from applying SIM to chatbots was that *suggestions* and *quick replies* could be used to help to deal with the conversational space associated with chatbots (see Chapter 4 - "Communicative Strategies Investigation"). That way, if a user does not know how to interact with the chatbot, they could pick one of the suggested *quick replies*, as opposed to a trial-and-error approach to discover the sentences the chatbot can understand. That approach, of course, comes at the cost of diminishing (or limiting) the possible communicative expressions users could utter and making the chatbot experience less like chatting and more like navigating a menu.

Motivated by these results, we decided to evaluate real users reactions while testing two chatbots with the same features but different ways of interaction: one focusing only on Natural Language Processing to define what the user wants, and the other making use of the sign classes (*quick replies*, *cards*, and *images*) to tell the user what they can do, not making use of NLP. Some of the strategies identified in Chapter 4 - "Communicative Strategies Investigation" were implemented in the bots to check how users perceive them (or not). Details about the chatbots and implemented strategies are further described below.

This chapter is organized as follows. First, we present both chatbots used in the tests in section 6.1 - "The Chatbots". Then, in section 6.2 - "The Tests", we explain the conducted tests. Next, on section 6.3 - "Analysis", we present our analysis and discussions for the obtained results. Finally, section 6.4 - "Conclusions" contains our conclusions.

6.1 The Chatbots

This section describes the chatbots that were used for the user tests: Kino and Cinemito. Both chatbots have the same context, which is cinema, and similar features for informing users about movies. Both chatbots were prototypes developed as case studies in the academic context, and cannot be considered finalized products. The main difference between them is their interaction paradigm. The objectives in developing the chatbots were also distinct: Kino was created to test the performance of NLP techniques, while Cinemito focused on implementing as many of the strategies from Chapter 4 - "Communicative Strategies Investigation" as possible. In the following subsections, we present each of them and the strategies they used.

6.1.1 Kino

The chatbot "Kino" was developed as part of a research of Natural Language Processing (NLP) techniques (D'Ávila, 2018). Kino was made as a rule-based approach for the design and evaluation of chatbots. Kino's features focus on information about movies, such as directors, producers, release date, among others. As Kino uses a rule-based approach, it matches the user input into one of the many rules programmed into it.

One of the objectives for developing Kino was to test three different rule-based techniques for NLP. For that, Kino would cycle between these three techniques when interpreting sentences. Later, D'Ávila (2018) analyzed which technique yielded the best results.

Although Kino's creator did not know our previous works, some of Chapter 4's strategies were implemented in Kino, as shown later in this section. Another important aspect of Kino is that its interface is completely textual and does not make use of any graphical sign classes such as *quick replies* or *cards* when interacting with users. That way we were able to explore the perception of these elements and their absence during the tests.

Kino uses "The Movie Database"'s API[1] (TMDB) as source of information, that way, it can answer questions about the movies present in TMDB's database. Kino uses Brazilian Portuguese as language, so users are required to input sentences in that language to talk to the chatbot. The only exception is the movie's title, which the users can refer to either by its Brazilian Portuguese version or its original (usually English) title. This happens thanks to the TMDB API search feature, which returns movies matching the input both as original title and Brazilian title.

The first message Kino sends when starting the conversation with users informs about its features and how to get help. Figure 6.1 on the next page depicts that message, Kino says "Hello [user name], I am Kino. I have just started to learn about movies. I can talk about directors, producers, release dates, running time, overview, and rating

[1] `https://www.themoviedb.org/`, last access on Jun/2020.

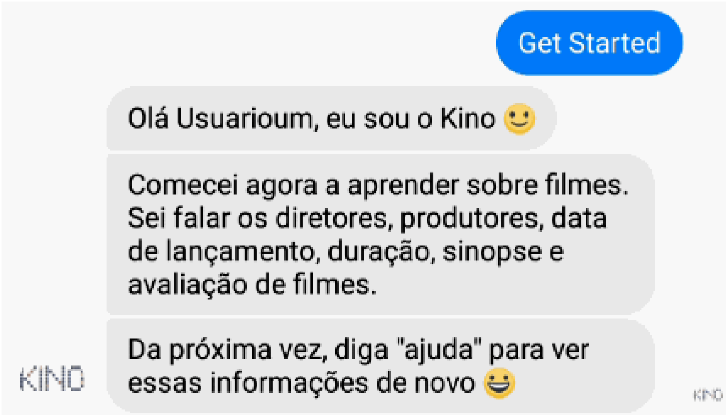

Figure 6.1: Kino's first message. Source: Authors
Translation: *–Get Started*
–Hello, Usuarioum, I am Kino :)
–I have just started to learn about movies. I can talk about directors, producers, release dates, running time, overview, and rating of movies. Next time, say "help" to see this information again :D

of movies. Next time, say "help" to see this information again". This first message is in accordance with the **S1** (*Showing the main feature on the first message*) strategy (described in Chapter 4 - "Communicative Strategies Investigation") and it also informs users how to access the help feature. Although very informative regarding what Kino can do, this message does not tell users *how* they can access the listed features.

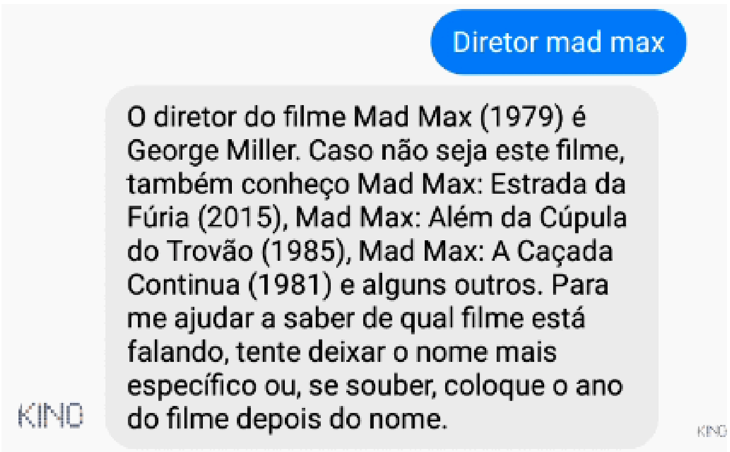

Figure 6.2: Kino chatbot informing the director of *Mad Max*. Source: Authors
Translation: *–Director mad max*
–The director of Mad Max (1979) is George Miller. If that was not the movie you were looking for, I also know Mad Max: Fury Road (2015), Mad Max: Beyond Thunderdome (1985), Mad Max 2 (1981), among others. To help me know which movie you are talking about, try typing its full name or, if you know it, type its release year after its name.

Interacting with Kino is very straightforward, users just have to type a question using natural language, and then the chatbot will answer (see Figure 6.2 on the preceding page). As a rule-based approach, the questions must follow a specific pattern so Kino can understand them. The pattern is first the desired information (director, producer, release date, running time, overview, and rating) followed by the movie title. For example, to know who directed *Pulp Fiction*, the user should ask something like *"Who directed Pulp Fiction?"* or other variations, as *"Who is the director of Pulp Fiction?"*, or even *"Director Pulp Fiction"*, as long as the pattern is respected. If the user asks in reverse order, e.g. *"Pulp Fiction, who directed it?"*, the chatbot will not understand the sentence.

In case of multiple movies with the same or similar names, Kino will reply the information asked and also a list of these movies' titles and the year they were released for disambiguation purposes, avoiding violating the Relation maxim of the Cooperative Principle (see Chapter 5 - "Pragmatics and Chatbots" for more information) in a similar way to the Wall Street Journal chatbot asking its user to clarify which company they want information about (see page 95 for that case). For example: if the user types *"Who directed Mad Max?"*, Kino will answer *"The director of Mad Max (1979) is George Miller. If that was not the movie you were looking for, I also know Mad Max: Fury Road (2015), Mad Max: Beyond Thunderdome (1985), Mad Max 2 (1981), among others. To help me know which movie you are talking about, try typing its full name or, if you know it, type its release year after its name."* (Figure 6.2 on the previous page), if the user was looking for the latest Mad Max movie, they should type something like *"Who directed Mad Max (2015)?"* or *"Who directed Mad Max Fury Road?"* to access the desired information. At the time of the tests, Kino did not keep the context of what was asked earlier. Because of that, if the same user, that just asked who directed *Mad Max: Fury Road*, wanted to know who produced the same movie and typed *"and who produced it?"*, Kino would display a message saying that it did not understand what the user had said. Instead, the user would have to type the name of the movie again, as in *"Who produced Mad Max (2015)?"*.

Besides messages regarding movies, Kino is also able to reply to some small talk, like "Hi", "Hello", "How are you?", and "Goodbye", as seen in Figure 6.3 on the following page. There is also a feedback rule to Kino, in which sentences such as "This answer is wrong", "I didn't like that answer", "I like it", or "thank you" are interpreted as feedback to a previously answered question.

Kino also has a *persistent menu* with two options, as seen in Figure 6.4. The first one is "What can I do?", which will trigger a response saying "I have just started to learn about movies. I can talk about directors, producers, release dates, running time, overview and evaluation of movies." (as shown in Figure 6.5 on page 115); and the second option is a shortcut for liking Kino's Facebook page ("Like our Facebook page"). When selecting that option, the user will be redirected to Kino's fan page on Facebook.

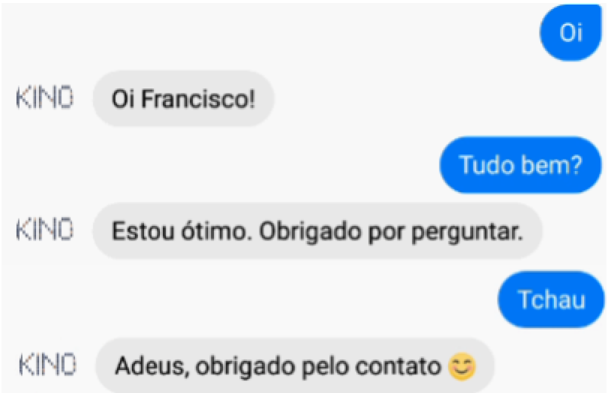

Figure 6.3: Kino chatbot answering to small talk. Source: Authors
Translation: –*Hi*
–*Hi Francisco!*
–*How are you?*
–*I'm great. Thanks for asking*
–*Bye*
–*Goodbye, thanks for contacting me :)*

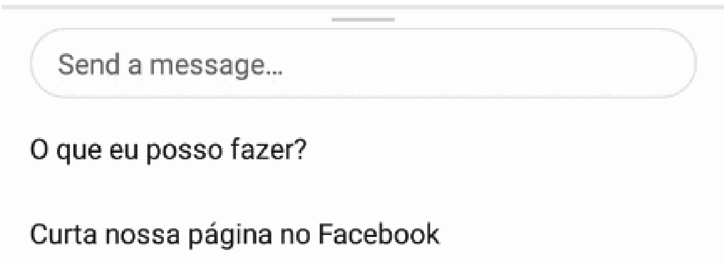

Figure 6.4: Kino's persistent menu. Source: Authors
Translation: –*What can I do?*
–*Like our Facebook page*

6.1.2 Cinemito

As we intended to consolidate the findings of Chapter 4 - "Communicative Strategies Investigation", mainly the identified strategies and sign classes under users' perspective, we decided to create a chatbot similar to Kino, with the same features (that is, informing users about movies' directors, producers, release date, running time, overview, and rating) but using a different approach for the interaction, focusing on the usage of other sign classes and strategies instead of NLP. The chatbot we have created is called "Cinemito".

Cinemito also answers users using Brazilian Portuguese, and it uses the same API as Kino as source of information, i.e. the TMDB API. Because of that, it is possible

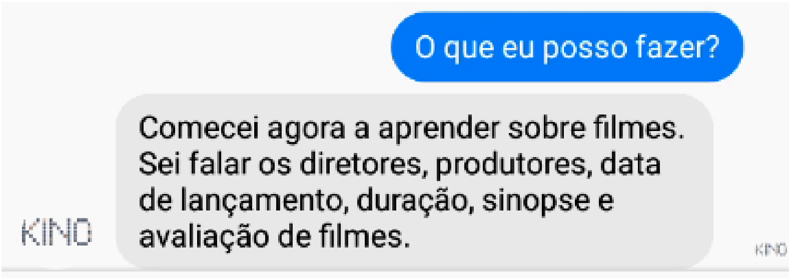

Figure 6.5: Kino's answer to "What can I do?". Source: Authors
Translation: *–What can I do?*
–I have just started to learn about movies. I can talk about directors, producers, release dates, running time, overview, and evaluation of movies.

to search for movies using either the original title or the Brazilian title. That way we managed to maintain consistency among the answers of both chatbots. Besides TMDB, we used Blip[2] as platform when creating Cinemito. We next describe how Cinemito works, indicating the design decisions based on the strategies presented in Chapter 4 - "Communicative Strategies Investigation".

Cinemito's first message informs users about its features, as proposed by strategy **S1** (*Showing the main feature on the first message*). Figure 6.6 on the following page shows that message: Cinemito says "Hello, [user name]! I am the chatbot Cinemito, I'm here to help you find information about movies using the TMDB database `www.themoviedb.org`. Do you want to learn how to use Cinemito?" and two *quick replies* are shown, for "Yes" and "No". This first message is intended to present the chatbot to users and also inform them about Cinemito's main feature and invite them to the tutorial on how to use it.

The short tutorial about how to use the chatbot is based on strategy **S2** (*Guiding the user through a short tutorial during first messages*). During this tutorial, users are asked to search for the movie "The Matrix", and then to use the offered *quick replies* to find out who directed it (Figure 6.7 on page 117 shows that tutorial). Users may, at any point, opt out of the tutorial. In that case, Cinemito will say it is OK to skip the tutorial, and that it will search for any movie title the user may type (as seen on Figure 6.8 on page 118).

We have designed Cinemito's interaction focusing on the search for the movie. Thus, whenever the user sends a message, the chatbot will search for a movie with that title and send a message repeating the exact query it is looking for, to let the user know exactly what was searched (as seen on Figure 6.9 on page 118). If one or more movies are found, the chatbot will say "I've found something" and a *carousel* with *cards* containing their *titles* and posters will be presented (Figure 6.10 on page 119 shows the *carousel* returned after searching for "Indiana Jones"). If the user selects the "*Mais informações*"

[2] `https://blip.ai/`, last access on Jun/2020.

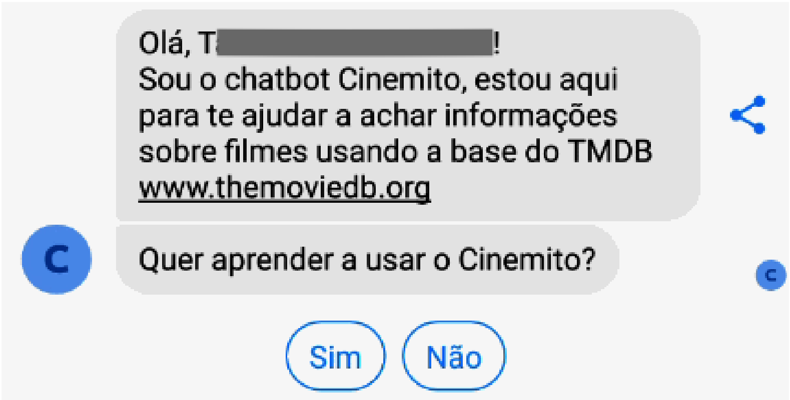

Figure 6.6: Cinemito's first message. Source: Authors
Translation: *–Hello, T–[Redacted name]! I am the chatbot Cinemito, I am here to help you find information about movies by using TMDB's database* `www.themoviedb.org`
– Do you want to learn how to use Cinemito?
-Yes -No

(more information) option in any of the presented movies, an image of the poster will be shown, along with its Brazilian title, its original title (as an example, Figure 6.11 on page 119 depicts *Raiders of the Lost Ark* poster) and a message telling users to choose which information they want to know about that movie (shown in the *quick replies* – one option for each available piece of information: director, producer, release date, running time, overview, and rating) or to type the title of another movie to search for it.

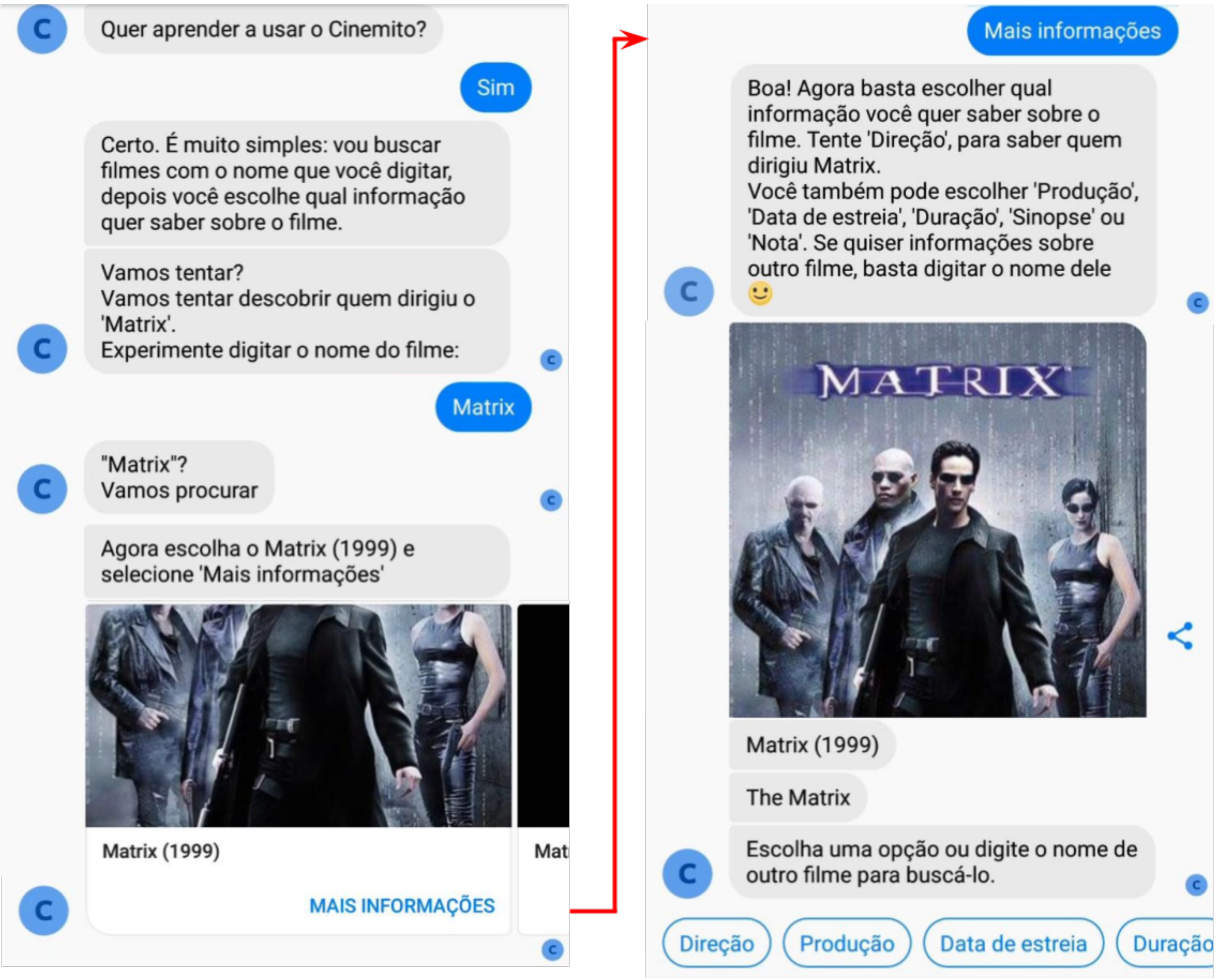

Figure 6.7: Cinemito's tutorial about its features. Source: Authors
Translation: *–Do you want to learn how to use Cinemito?*
–Yes
–Right. It is very simple: I'm going to search for movies with the title you type, then you choose which information you want to know about that movie.
–Let's try? Let's try to find out how directed "The Matrix". Try to type the title of that movie:
–Matrix
–"Matrix"? Let's search for it
–Now choose Matrix (1999) and select "More information"
[on right hand of the figure] *–More information*
–Great! Now you just have to choose which information you want to know about the movie. Try "Direction", to know who directed The Matrix. You can also choose "Production", "Release date", "Running time", "Overview", or "Rating". If you want information about another movie, just type its title :)
–Choose an option or type the name of another movie to search for it.
-Director -Producer -Release date -Running time

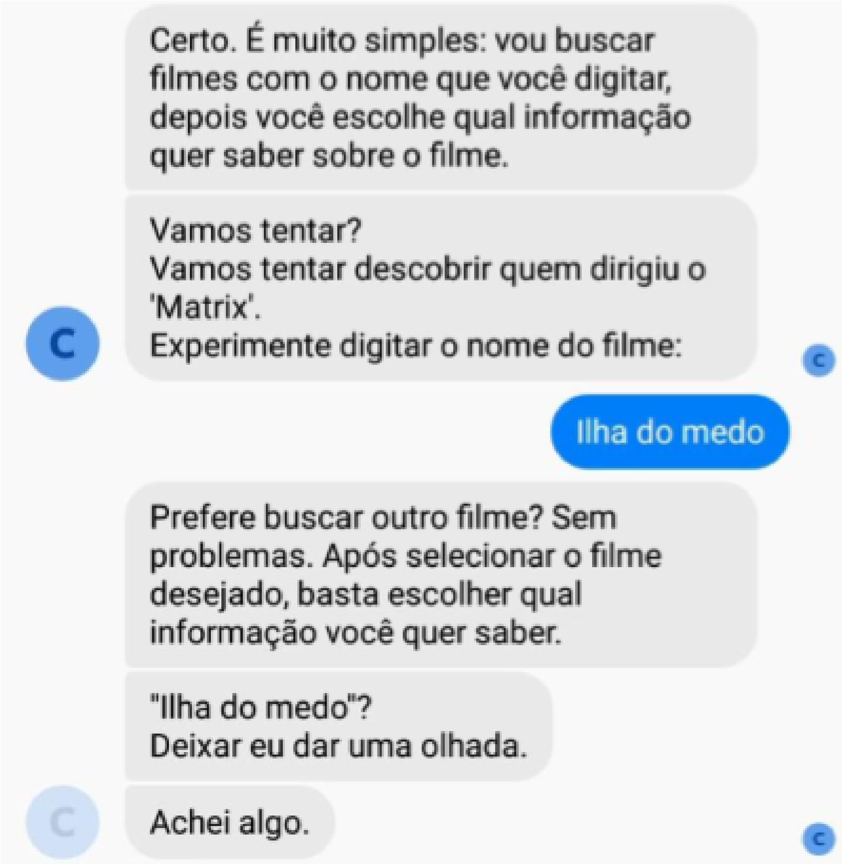

Figure 6.8: User opting out of Cinemito's tutorial. Source: Authors
Translation: –*Right. It is very simple: I'm going to search for movies with the title you type, then you choose which information you want to know about that movie.*
–*Let's try? Let's try to find out how directed "The Matrix". Try to type the title of that movie:*
–*Shutter Island*
–*You rather search for another movie? No problem. After selecting the movie you want, you just have to choose which information you want to know about.*
–*"Shutter Island"? Let me take a look.*
–*Found something.*

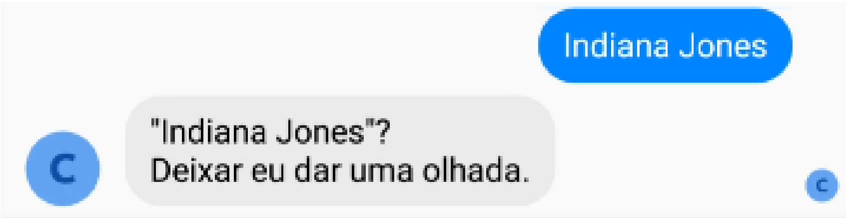

Figure 6.9: Cinemito searching for a movie called "Indiana Jones". Source: Authors
Translation: –*Indiana Jones*
–*"Indiana Jones"? Let me take a look.*

Figure 6.10: Cinemito showing a *carousel* of some of the movies returned after searching for “Indiana Jones”. Source: Authors

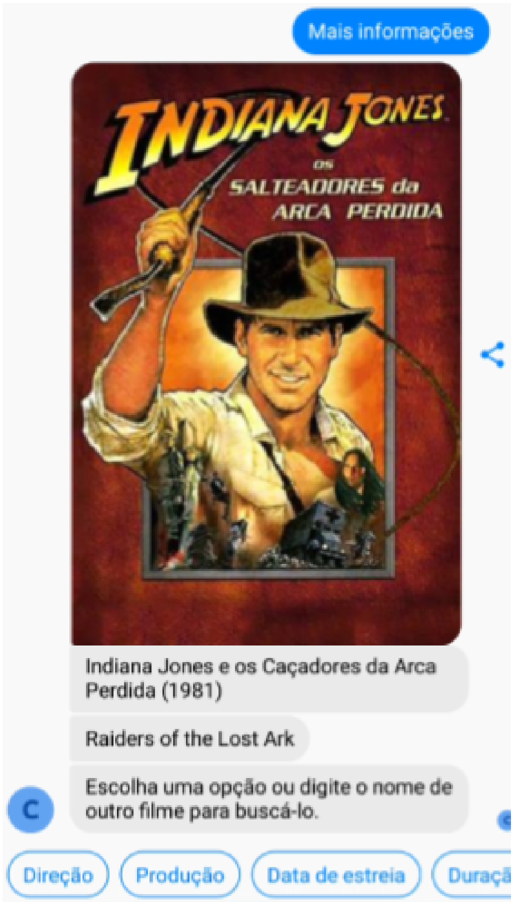

Figure 6.11: Cinemito showing the poster for Indiana Jones Raiders of the Lost Ark. Source: Authors
Translation: –*More information*
–*Indiana Jones e os Caçadores da Arca Perdida (1981)* [Brazilian title]
–*Raiders of the Lost Ark*
–*Choose an option or type the name of another movie to search it.*
-Director -Producer -Release date -Running time

The *quick-replies* follow strategy **S3** (*Suggesting the next possible set of actions to the user*) and suggest to users other pieces of information about the movie they may be interested in. That way the user is reminded (or made aware) of which other information about the movie Cinemito can provide them, and also is spared of typing the movie title again. If users select a *quick reply*, the chatbot replies with the requested information, and then the list of *quick replies* is shown again, in case the user wants to know more about that movie. That way the user does not need to type the movie title for each desired information.

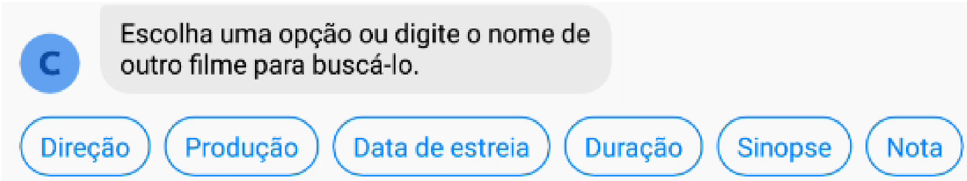

Figure 6.12: Cinemito showing *quick replies* the user may choose from. Source: Authors
Translation: *–Choose an option or type the name of another movie to search for it.*
-Director -Producer -Release date -Running time -Overview -Rating

Our design solution considers that the title of the movie is the main topic of the conversation, thus anytime users type something, Cinemito will search for a movie with the typed string as a title. Although this solution may make it easier for users to have information about movies, it has an associated cost: Cinemito does not answer any other kinds of questions or small talk from users. Thus, if a user asks "Indiana Jones director", Cinemito will search for a movie with that exact name, causing a communication breakdown in the conversation, as seen in Figure 6.13 on the following page. To avoid that kind of situation, Cinemito will always repeat what it is going to be searched using quotes, so that the user may realize that the chatbot is only searching for those exact terms. Figure 6.13 on the next page shows Cinemito repeating the query ""Indiana Jones director"? Let me take a look." before replying "I have not found any movie with that title :/ Try using other words.".

That is also the case for **S8** (*Showing the main menu or the most frequent features when user asks for help*), when users ask for help: if the user types "*Ajuda*" ("help" in Portuguese), the chatbot will search for movies with that name. To get help, the user is supposed to select the "*Ajuda*" option in the *persistent menu*. Cinemito's *persistent menu* is based on strategies **S4** (*Having a persistent menu with main features*) and **S10** (*Showing the persistent menu instead of a text-input box*). However, due to our design approach, it does not contain the chatbot's main functions, instead, it shows options that cannot be accessed from the chat interface, such as "Help", "About", and a sub-menu called "More..." with options for "Restart conversation" and a link to the "TMDB's site", as shown on Figure 6.14 on the following page.

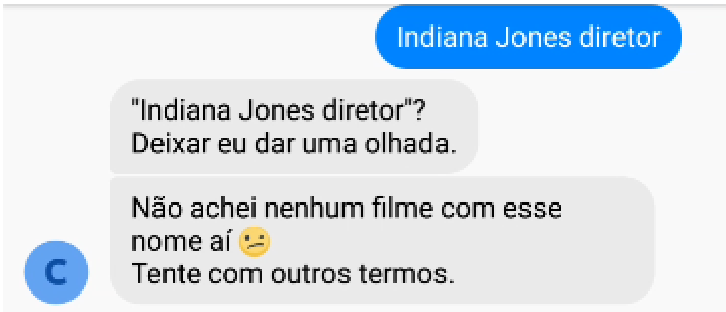

Figure 6.13: Communicative breakdown when trying to use Cinemito. Source: Authors
Translation: –*Indiana Jones director*
–*"Indiana Jones director"? Let me take a look.*
–*I have not found any movie with that title. Try using other words.*

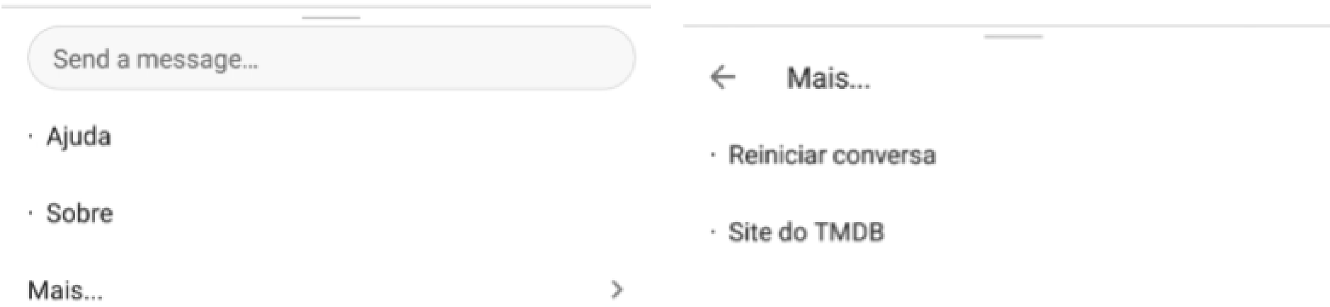

Figure 6.14: Cinemito's *persistent menu* (left) and sub-menu (right). Source: Authors
Translation: -*Help* -*About* -*More...* [on the left-hand side of figure]
-*Restart conversation* -*TMDB's site* [on the right-hand side of figure]

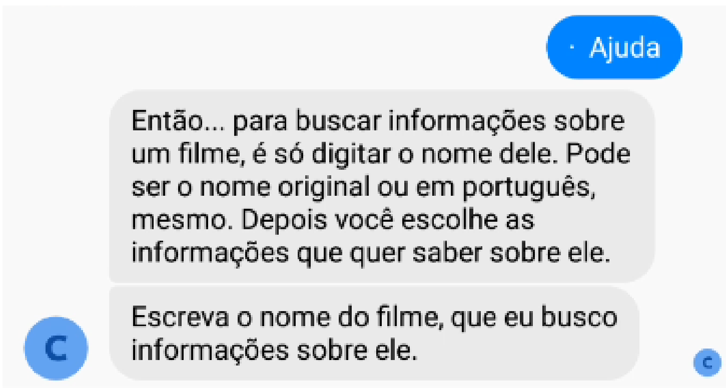

Figure 6.15: Help message from Cinemito. Source: Authors
Translation: –*Help*
–*So... to look for information about a movie, you just have to type its title. It can be its original or translated title. Then, you choose which information you want to know about it.*
–*Write the movie title and I will search for information about it.*

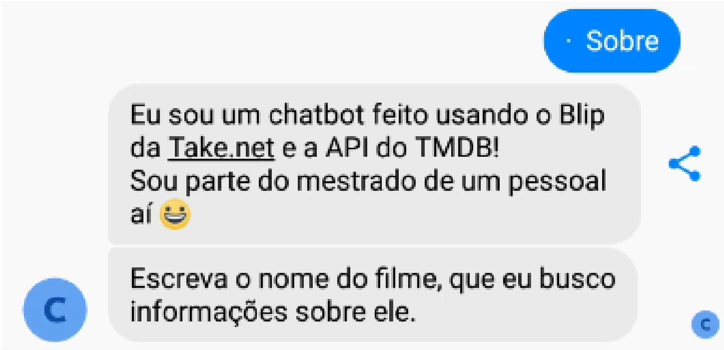

Figure 6.16: Cinemito's "about" message. Source: Authors
Translation: *–About*
–I am a chatbot that was mad using Take.net's Blip and TMDB's API! I am part of the masters of some people :)
–Type the title of a movie that I will search for information about it.

The "Help" option will prompt the chatbot to inform about how to use it: "So... to look for information about a movie, you just have to type its title. It can be its original or translated title. Then, you choose which information you want to know about it. Write the movie title and I will search for information about it.", as shown in Figure 6.15 on the previous page. "About" will trigger a message informing that the chatbot was built using Take.net's Blip and TMDB's API as part of a master's thesis (Figure 6.16). Finally, the sub-menu lists a link that will open TMDB's site and the option to restart the whole conversation. The restart conversation option was built to help to test the chatbot while developing it, but it can also be used by users that want to reset their conversation and see the chatbot's first messages again.

6.1.3 Strategies and Sign Classes Usage

In this section, we compare Kino and Cinemito's interaction with users based on the sign classes and strategies presented in Chapter 4 - "Communicative Strategies Investigation". For easier comparison, Table 6.1 on the next page shows which sign classes were used in each of the chatbots (Kino and Cinemito). For each line, if a sign class is present in the chatbot, a "✓" is marked; a "." marks otherwise. Kino made use of *simple message* and *persistent menu* sign classes; Cinemito used all of the six sign classes.

Table 6.1: Sign classes usage on Kino and Cinemito

	Kino	**Cinemito**
Simple message	✓	✓
Simple image	.	✓
Quick reply	.	✓
Card	.	✓
Carousel	.	✓
Persistent Menu	✓	✓

Regarding the strategies for presenting the chatbot's features, Table 6.2 on the following page shows their usage on both Kino and Cinemito. Each line in the Table represents a strategy, and the cells are marked with a "✓" indicating the presence of that strategy on the chatbot, and a "." indicates otherwise.

Kino's designer was not aware of the strategies we compiled in the first part of this work when he created his chatbot. Even so, we could identify five of them in Kino. As we said before, Kino shows its main features in its first message (**S1**); it also has a *persistent menu* (**S4**); shows its features when the user asks for help (**S8**); shows its features when it cannot understand what the user said (**S9**); and, when used on a mobile device, it shows the *persistent menu* instead of a text-input area (**S10**).

For Cinemito, we sought to include as many of the strategies as the chatbot's features and interaction style allowed us, as we intended to verify the users' opinions about these strategies. That way we implemented strategies **S1** (*Showing the main feature on the first message*), **S2** (*Guiding the user through a short tutorial during first messages*), **S3** (*Suggesting the next possible set of actions to the user*), **S4** (*Having a persistent menu with main features*), **S8** (*Showing the main menu or the most frequent features when user asks for help*), and **S10** (*Showing the persistent menu instead of a text-input box*) on Cinemito. We were not able to use strategies **S5** (*Having a main menu with main features*), **S6** (*Having a list of available commands*), **S7** (*Offering contextual help about a feature*), and **S9** (*Showing the main menu or the most frequent features when user asks for help*), as the context is very simple and we opted to make Cinemito's interaction based on *quick replies*, considering everything users type as a movie title to be searched; **S11** (*Highlighting the most important features*) was also not implemented, as Cinemito has only a single feature.

Table 6.2: Strategies usage on Kino and Cinemito

	Kino	Cinemito
S1 (*Main feature on first message*)	.*	✓
S2 (*Tutorial on first messages*)	.	✓
S3 (*Suggesting possible actions*)	.	✓
S4 (*Having a persistent menu*)	✓	✓
S5 (*Having a main menu*)	.	.
S6 (*Having a list of commands*)	.	.
S7 (*Having contextual help*)	.	.
S8 (*Showing the menu/features when user asks for help*)	✓	✓
S9 (*Showing the menu/features when the bot cannot understand*)	✓	.
S10 (*Showing the menu instead of a text-input on mobile*)	✓	✓
S11 (*Highlighting most important features*)	.	.

* Although Kino had strategy **S1** implemented, its first message was not displayed during tests. So for analysis purposes, it was considered as not using **S1**.

6.2 The Tests

The user tests aimed to check users' preferences and opinions regarding the usage (or not) of the strategies from Chapter 4 - "Communicative Strategies Investigation" and different ways of interacting with chatbots having similar features. The gist of it was to give a task to the user and ask them to use one of the chatbots to do it, later we would have them perform another similar task using the other chatbot, finally, we would conduct a semi-structured interview to understand their actions during the experiment. To avoid learning effects, we counterbalanced the chatbot order throughout the tests, that is, half the users used Kino first, while the other half used Cinemito in their initial task.

In preparation for the tests, we created a scenario that would guide the user interaction with the chatbots, so they would know what to look for when using the chatbot. The scenario we used was the following:

> "*You and your friends are in a pub having fun and chatting. Eventually, the group starts discussing movies, Indiana Jones movies, in particular. But no one seems to agree about who directed the first Indiana Jones movie. One of your friends is saying that George Lucas was the director, as the franchise was part of the LucasFilm. But another friend is saying that is not true. You do not know which of them is correct (if any), but you just remembered about a chatbot that can answer questions about movies. And then you decide to end that discussion once for all and check whether George Lucas directed the first Indiana Jones movie.*"

First of all, we chose Indiana Jones because it is a very popular and well-known movie. The scenario was written to impose some challenges to the user that would help us to discover how the features of the chatbots were perceived by them. George Lucas is indeed a name very associated with Indiana Jones, but he is not the director of the movie. That way, as a follow-up question we could ask if he was the producer of those movies. Besides that, Indiana Jones is a series with many movies, but none of them is numbered. There is no "Indiana Jones 2", rather it is titled "Temple of Doom", the first movie in the series is also not called "Indiana Jones", but "Raiders of the Lost Ark". That way, users during the test would have to discover which Indiana Jones movie was the first one.

A variation of that scenario was also used, just changing the movie series to Harry Potter. For example, asking who was the producer of the fourth Harry Potter movie, as the movies in that franchise are also not numbered. Another follow-up task used during the tests was to ask the user to check if the director of the last "Mad Max" movie was the same as the first one.

The variation scenario (using Harry Potter) and the follow-up tasks were only used as a sequence to the main scenario in cases in which the user had completed the main task and wanted to keep interacting with the chatbot for a while. That way, for each chatbot, users would complete the main scenario (Indiana Jones) and one other follow-up scenario, except in one case in which the user did not want to keep interacting with the chatbot after the main scenario.

We also changed the order of the chatbots in each test as a strategy to mitigate the learning bias effect, that way, five users tested Kino first, while the other six started their tests using Cinemito. But, as they used the same scenarios, users may have learned the complete title of the movie when testing the second chatbot.

As part of the evaluation, we conducted a semi-structured interview with each participant. The script for the interviews is composed of three parts: first, some demographic questions composing the pre-test interview; then, the post-test interview with questions about the first chatbot used in the test; then questions about the second chatbot, also comparing it to the first one; finally, questions about some aspects of the chatbots (related to the strategies) the users may have noticed (or not). Note that questions asking to score the chatbot's characteristics are on a scale of 1 to 5, and users were also encouraged to further explain all of their answers. The final script is shown next:

Table 6.3: Interview script

Block	**Main items**
1. Demographics	a) Age b) Education: i. Courses ii. University c) Movies: a) Appreciation b) Frequency
2. After testing the first chatbot	a) Scale 1 (lowest) to 5 (highest): i. Ease of use; Justify ii. Intelligence; Justify iii. Would recommend it? Justify b) What caught your attention? c) Positive points d) Negative points e) Enjoyed using it? f) Would use it again? g) Suggestions h) Comments

Table 6.3: Interview script, continued

Block	Main items
3. After testing the second chatbot and comparison:	a) Scale 1 (lowest) to 5 (highest): i. Ease of use; Justify ii. Intelligence; Justify iii. Would recommend it? Justify b) Which one you enjoyed most? Why? c) Which one seems more complete? Why? d) Which one seems more intelligent? Why? e) Which one is easier to use? Why? f) Both chatbots understand sentences equally well? i. Does any of them not use NLP? ii. What do you think of that? g) Previous experience with chatbots h) These chatbots against the ones you knew

Table 6.3: Interview script, continued

Block	Main items
4. Strategies and Sign Classes:	a) Noticed a message informing what the chatbot can do? What do you think about that? (**S1**) b) Noticed the tutorial? What do you think about that? (**S2**) c) Noticed the *persistent menu*? What do you think about that? (**S4**) d) Noticed the help option? What do you think about that? e) Noticed suggestions of replies? What do you think about that? (**S3**) f) Noticed the *carousel* of movies? What do you think about that? g) Noticed the feedback feature? What do you think about that?

Before conducting the tests, we performed a pilot test to verify whether the format of the test and the script were adequate or needed any adjustments. That pilot test happened without any problems. The only modification after the pilot was changing the expected duration of the test from 20 to 40 minutes.

The participants were recruited based on convenience. The eleven participants were University students, most of them having a technology background. Although not all of them were familiarized with chatbots, all of them had used conversational interfaces at least once before, especially Siri.

Out of the eleven participants, two were female, while the other nine were male. All of them came from the academic background: eight of them have studied or are studying courses related to the Computing field, one came from another STEM area, and two of them had undergrad degrees in humanities areas. Even so, all of them knew how to program and had familiarity with smartphones and Facebook Messenger. Out of the volunteers, two of them were pursuing their undergrad degree; four were pursuing MSc degree; and five were Ph.D. candidates. Their age varied from 20 to 33 years.

Ten of the participants said they enjoy movies and one said they "sort of" liked it. That shows that all of them were familiar with movies, and were likely to have no problems understanding the scenario and chatbots' context. Eight of them said they watched 3 to 5 movies per month; one of them watched up to ten movies per month; and, on the other end, two people said they watched two or fewer movies per month.

Table 6.4 shows an overview of participants demographics, such as their age, gender, ongoing academic degree, among others. The second column (First test) shows which chatbot each participant tested first: "C" indicates Cinemito and "K" means Kino. Participant *P3* is marked with an asterisk because *P3* only interacted with Kino, as technical difficulties prevented the realization of the Cinemito test, and the results of his test were only considered in the analysis of Kino, not during the comparison between chatbots.

The tests took place from May 7^{th} 2018 through May 10^{th} 2018. During the tests, both chatbots were accessed through Facebook Messenger. We figured that it would be a familiar environment for most users, and that way they would not be affected by an unacquainted piece of software. The smartphone used in the tests was an Android phone with a 5.5-inch screen, model Oneplus 3T. The smartphone also ran the software for recording the audio and capturing what was shown on the screen.

The experiments occurred in controlled lab space at the university campus. During the tests, only the participant and the researcher were present in the room. The researcher would then inform the users that what was being evaluated was the different ways of interacting with chatbots, not the users themselves. The researcher would also tell the user about the consent form, that they were volunteers, and that at any moment the user could stop the test without any prejudice to them, and that the session would be recorded and further analyzed for research purposes.

The researcher would then start recording the session and initiate the experiment by asking the user questions about demographics described in the script. After that, the researcher would read the scenario to the user and then hand them a smartphone with

Table 6.4: Participants demographics

	First test	Age (years)	Gender	Academic Degree (ongoing)	Familiar with FB messenger?	Likes movies?	Movies watched per month
P1	K	27	Male	PhD	Yes	Yes	3-4
P2	C	32	Female	PhD	Yes	Yes	4
*P3**	K	25	Male	PhD	Yes	Yes	4
P4	C	33	Female	MSc	Yes	Yes	3-4
P5	C	28	Male	PhD	Yes	Yes	10
P6	K	27	Male	MSc	Yes	Yes	5
P7	C	22	Male	Undergrad	Yes	Yes	3
P8	K	29	Male	PhD	Yes	Yes	4-5
P9	C	26	Male	MSc	Yes	Yes	2
P10	K	24	Male	MSc	Yes	Sort of	1 at most
P11	C	20	Male	Undergrad	Yes	Yes	4-5

the chatbot already running. During the tests, users were asked to apply the think-aloud protocol (Lewis, 1982), that is, they were asked to speak out loud (or narrate) what they were thinking as they were interacting with the chatbot. After the user finished the scenario, the researcher would read the follow-up scenario and allow them to keep interacting with the chatbot until they were satisfied (most users kept using the chatbot for another five minutes). Then, the researcher would interview them about what they thought of the chatbot.

After the interview, the researcher would read the scenario again and set the second chatbot on the smartphone. When the user finished the scenario, they would be given an optional follow-up scenario and would be allowed to keep exploring the second chatbot for as long as they wanted (again, most users kept using it for another five minutes). After that, a few more questions were asked, first regarding the user's opinion of the second chatbot, and later comparing both chatbots. Finally, the researcher would ask questions regarding the user's perception and opinion about some of the chatbot's aspects that were not directly explored in the scenario.

Unfortunately, despite our best efforts, we had two setbacks. First, the introduction message from Kino was not shown to the users due to server problems. That situation happened on all of the tests except the pilot one. Because of that, we considered that the results would still be valid, as all of the tests had experienced the same problem. The second problem we ran into was during the second part of one of the tests: we had severe connection problems and, because of that, the participant was not able to use the second chatbot (Cinemito). For that reason, the results of this test were only considered for the Kino analysis, (and not considered when analyzing the third part of the script – questions about the second chatbot and comparison between both chatbots).

6.3 Analysis

The audio and smartphone screen were recorded during the interviews, and the interviewer took notes about interesting points about the participant interaction with the chatbots, as well as comments they made due to the think-aloud technique. During the analysis, the recordings and notes were compiled and grouped according to recurring themes. That way, mistakes that happened to more than one participant were highlighted and their possible causes could be investigated through participants' comments and answers (e.g. trying to ask a complete question to Cinemito before realizing that they should just type the movie title). Similar recurring themes on participants' answers to the script questions were also grouped during the analysis. That enabled us to compile a list of good and bad characteristics of the chatbots according to the participants. Finally, interesting outliers were also highlighted during the analysis.

This section presents the results of the analysis of the tests and interviews we conducted as part of this work. The results are presented based on the script used during the interviews. They show the main results as well as any interesting outliers that happened during the tests.

6.3.1 Comments about Kino

In this subsection, we are grouping what was said about Kino. This includes comments both from the five users that tested it first and the six other users who tested it last, after using Cinemito.

For these first three questions, we asked all eleven users to use a scale with values from 1 (lowest) to 5 (highest) to qualify some aspects of the Kino chatbot. Users were also asked to justify their answers. The same questions were also asked about the Cinemito chatbot and are presented in the next subsection. These questions refer to subsections 2(a) and 3(a) of the script.

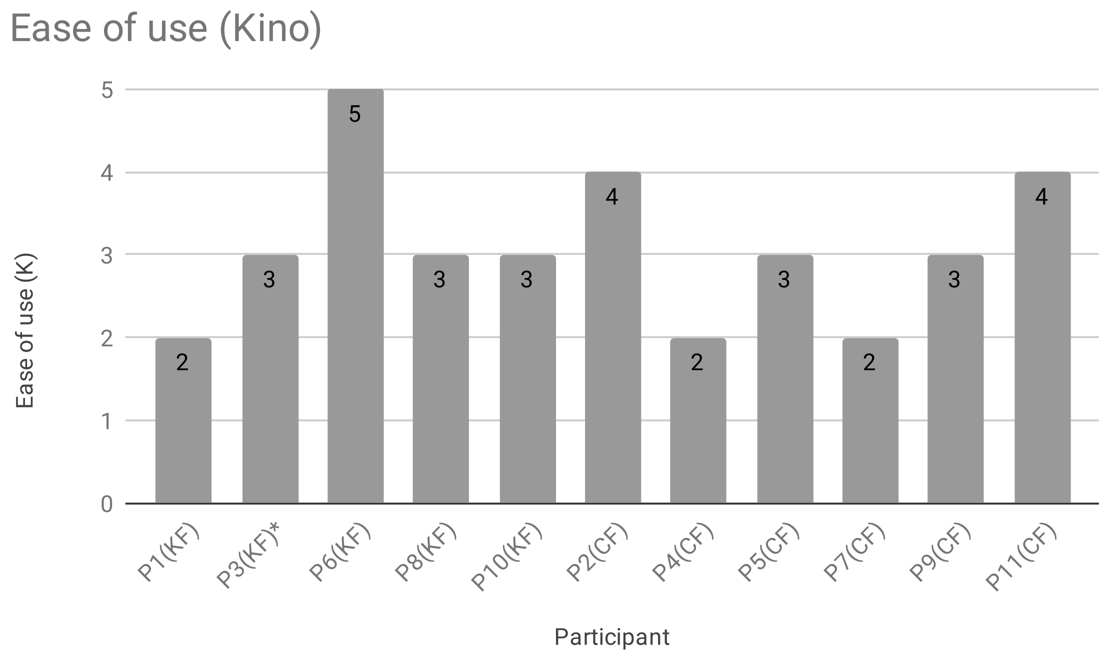

Figure 6.17: Scores for Kino's ease of use. Source: Authors

Regarding **ease of use**, the mean of the scores was 3.09, with a minimum of 2 and a maximum of 5. Figure 6.17 on the preceding page shows the scores for each participant and indicates whether the participant tested Kino first (the first five columns, marked as KF) or not (other columns, marked as CF). Users that gave a low score complained that the chatbot could not understand some commands, nor could answer every question; that it could only understand a very specific command and, sometimes, it did not understand the title of a movie. Other complaints focused on the lack of information regarding the pattern of the messages the chatbot can understand (in this case, users should follow the pattern <desired information> <movie title>). Because of that, users relied on trial and error until they discovered how to interact with the chatbot. Three users (P4, P5, and P8) also said that they did not know what they should do or type to the chatbot and that the "help" message was not clear and did not help. P5 also did not know whether he should use question marks or not, and that it was confusing. When starting the interaction, P4 said: "How can I know what you can do?". Some of these results may have been influenced by the lack of Kino 's initial message explaining how to use the chatbot due to a problem with Kino during all tests. Even so, some users could not understand how to use Kino even after reading the help feature's message. P6 was the only participant that gave Kino the highest possible score for ease of use: he complained about the unclear help message but was able to finish the task very quickly, so he justified the score saying that the chatbot was easy to use, just like chatting.

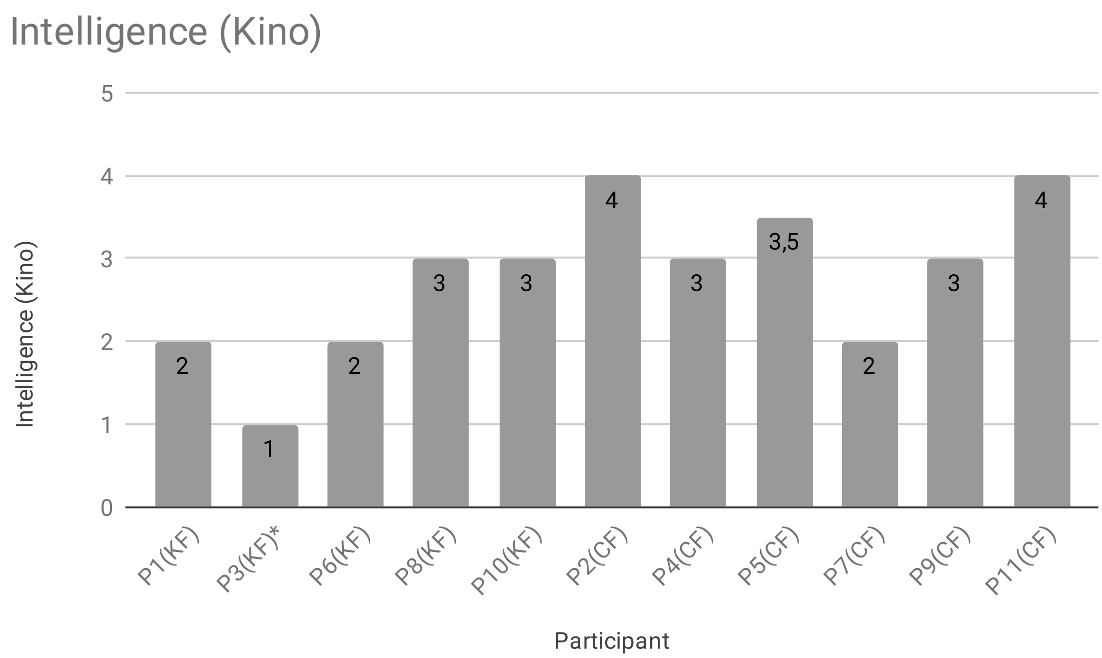

Figure 6.18: Scores for Kino's intelligence. Source: Authors

About how **intelligent** Kino was, the mean score was 2.77, with a low of 1 and a high of 4, Figure 6.18 on the previous page shows individual scores, the first five columns show scores from users that tested Kino first (marked as KF), while the other columns show scores from participants that tested Kino after testing Cinemito (marked as CF). In general, the perception was worse than that of ease of use. The main comments were about the chatbot sometimes not understanding the title of a movie, and users having to ask the chatbot in a very specific manner for it to understand (users have to use the pattern *<desired information>* *<movie title>* so Kino can understand the sentence). P3, P8, and P9 said that the chatbot should allow searching for a movie just by typing its title instead of having to type the whole question, especially when asking for other information about the same movie as before, as P9 said: “I have to type it all over again”. P9 also complained that Kino did not keep the context of the conversation. P5 said the information was presented confusingly. P11 mentioned that Kino had information about a lot of movies, but it did not have information for unreleased ones. In general, the comments were harsher than the actual scores.

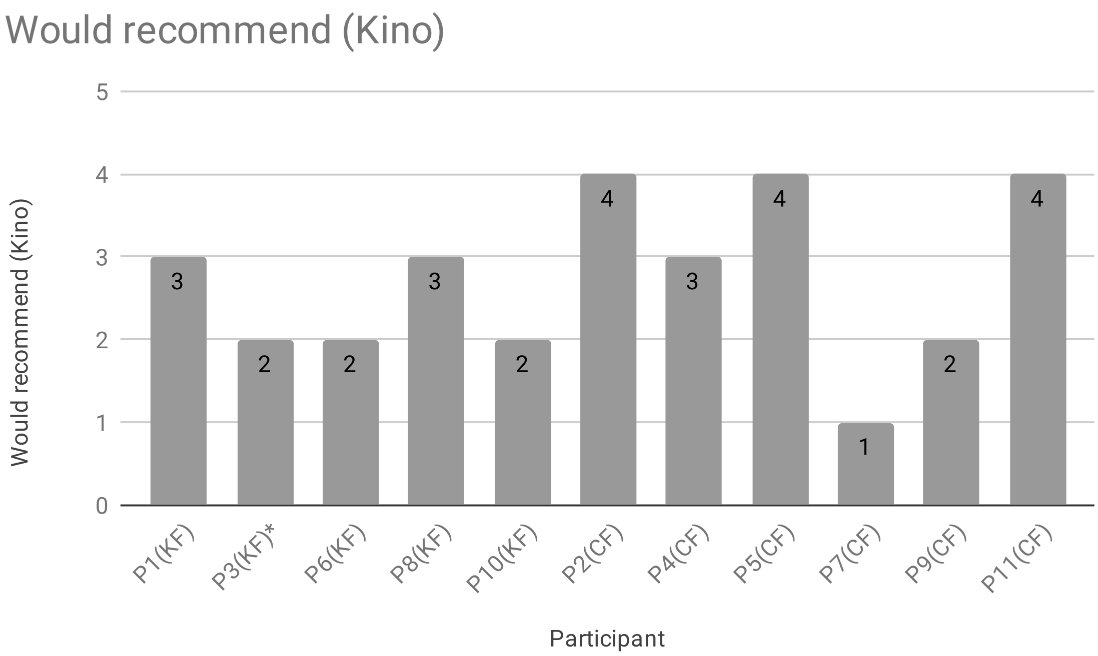

Figure 6.19: Scores for how likely it is to recommend Kino to someone. Source: Authors

When asked if they would **recommend** Kino to other people, the mean answer was 2.73, with one 1 as minimum, and 4 as maximum. Figure 6.19 shows these scores, the first five columns have scores from participants that had Kino as the first test (KF), while the others have scores from those that tested Cinemito first (CF). These scores were also lower than ease of use’s. P11 pointed that Kino is nice, but it would not improve anyone's productivity. P2 said that as it is a conversation, it is nicer than just searching, but she also said, along with P3 and P7 that it takes too long to use the chatbot to get the information. P1, P7, and P11 also mentioned that using Google would be easier and faster. P6, P9, and P10 said that it did not understand every question. P4, P7, and P9 also said that it was not very easy to use, and P4 would not recommend it for someone inexperienced with tech. P5 said: "...my mother would not be able to use this one".

* * *

As we said before, the previous questions about Kino were answered by all of the eleven participants, including five users that tested Kino first and also the six others that tested it last.

The next questions in this subsection refer to subsections 2(b) to 2(h) of the interview script and were only asked to the five users that tested Kino first, right after the test and before using the other chatbot.

What **caught the attention** of users testing Kino was: its name (P1), the fact that it can answer a few questions (P1), the fast response time (P6), that Kino suggests movies with similar names (P10). Participant P3 praised its aesthetics, but the user was referring to Facebook Messenger's interface. P8 complained that sometimes Kino did not understand a very popular movie title ("When I put a very popular title, I expect it to find something"), and also noted that using natural language for interacting with software feels strange, but maybe using voice commands it would feel better.

As **positive points** about Kino, users explained that it is friendly (P1), fast (P6), very thorough when it understands the question (P1, P8, and P10). Participant P8 also complimented its politeness, as Kino apologized when there were errors, taking away the blame from the user ("It says it's sorry about that [the error], taking the blame away from me"). P3 and P10 said that some of the help messages were useful.

The **negative points** users indicated about Kino were that it could not understand questions that do not follow the accepted pattern (P1, P6, P10). For example, P10 tried asking "Was George Lucas the director of Indiana Jones?" and some variants before noticing that he should ask "director of Indiana Jones". P6 also did not like the quality of the answers. P8 said that the chatbot should search for information about the movie directly when he just typed its title ("It should answer something right away when it's something famous. If I want to know something about Titanic, I would type just that").

All five users said they **enjoyed using** Kino, although P8 said "I've enjoyed it more than I did not enjoy it", indicating mixed feelings. P6 also said it still could be improved, as the technology for doing so already exists. Participant P10 enjoyed trying to figure out what Kino was able to do.

But when asked whether they **would use it again**, only participant P3 said he would. P10 said he would "probably use it, if necessary". Participant P6 said he might consider using an improved version.

The **suggestions** the participants made focused mainly on improving the NLP (Natural Language Processing) of the chatbot (P1 and P8), as well as showing to users some examples of sentences the chatbot would understand (P3 and P10). Participants P6 and P8 would like to see images and posters of the movies, P6 would like to see more information about the movies, such as plot spoilers.

Only participant P6 wanted to make further **comments** on the chatbot and suggested that Kino could show posters of the movies and have links to their website.

6.3.2 Comments about Cinemito

This subsection groups the comments made about the Cinemito chatbot. We are grouping the answers of the users that tested Cinemito first, as well as those that used Cinemito last.

The first questions were answered by ten participants, including four[3] that tested Kino first, and the other six participants that tested Cinemito first. In these questions, we asked users to use a scale of 1 (lowest) to 5 (highest) to score some aspects of Cinemito and then justify their scores. The questions refer to subsection 2(a) and 3(a) of the script.

The mean score the users gave to Cinemito's **ease of use** was 4.4. The minimum score was 3, and the maximum was 5. Figure 6.20 shows these scores: the first five columns marked with KF have the scores by the participants that tested Kino first (P3 did not test Cinemito, so the column for his score is empty), while the other six columns indicate the scores of those who tested Cinemito first (CF). P6, one of the participants who gave the lowest scores, mentioned that Cinemito seems like a simple query to a database. P2 complained about uninformative error messages and that it took some time for her to realize how to use the chatbot, as she was trying to ask questions instead of just typing the movie title, as she said: "It's a chatbot with no chatting". P4 and P7 users praised the tutorial, saying it helped by telling beforehand how to use the chatbot. Participant P8 praised how the chatbot made the instructions of how to use it clear right away. Five participants (P4, P6, P7, P9, and P10) also complimented the use of *quick reply* bubbles, P4, P7, and P10 also said the bubbles made clear what the chatbot could do. P10 said he does not like to type, and the *quick replies* help with that, so he does

[3]Due to connection problems, P3 did not test Cinemito, so this part of the analysis does not include this interview.

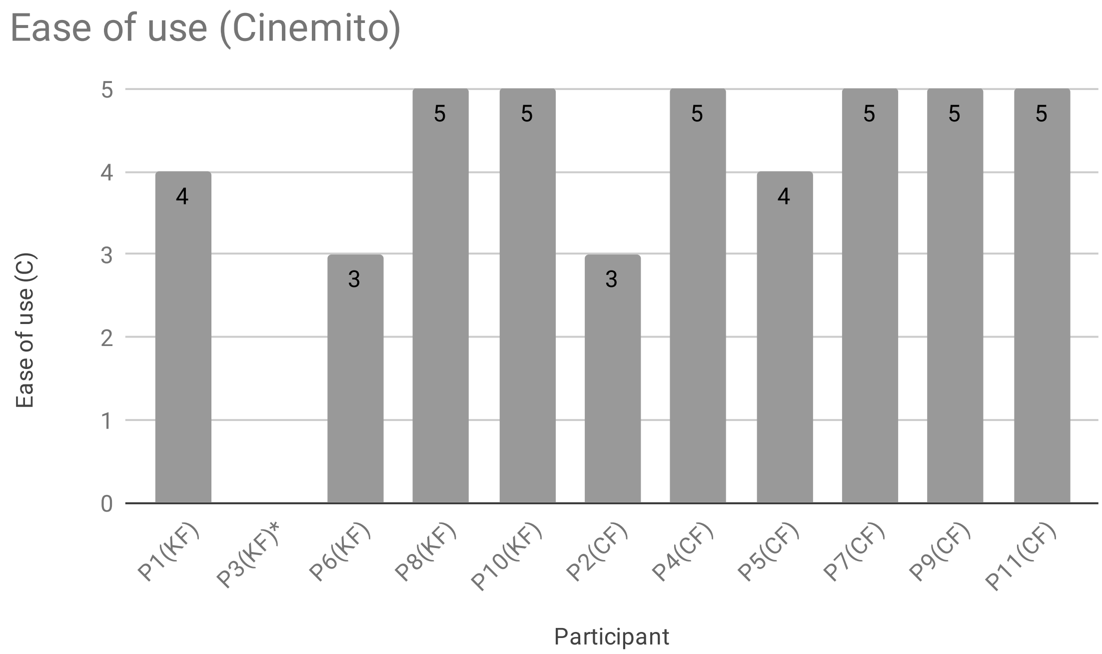

Figure 6.20: Scores for Cinemito's ease of use. Source: Authors

not have to type a lot. P8 noted that using *quick replies* is mandatory, and it would be nice if Cinemito would also work when typing the requested information instead of just by selecting the *quick reply*. P8 also praised that the chatbot never let him wondering what he was supposed to: "I didn't have any dead ends [during interaction]".

Regarding users' perception of Cinemito's **intelligence**, the mean score was 3.1, with a low of 1 and a high of 5. These values are shown in Figure 6.21, in which the first five columns[4] contain scores from those who used Kino first (KF), while the other ones have the scores from participants that used Cinemito first (CF). This result was higher than expected, as Cinemito makes no use of NLP techniques and only tries to search for a movie with whatever the user types as the title. After checking their justifications, we could see that P2, which gave a score of 1, explained that Cinemito simply makes pattern matching. P6 and P8 commented that Cinemito is very limited and does not try to hide it but it is efficient at what it can do. P4 pointed that Cinemito has a lot of information about movies. P1 and P9 said that it cannot answer simple sentences like "Hi" and "Good morning" nor any questions. P10 noted that it can also search for movies with their original titles. P5, which gave the highest score, explained that the chatbot seemed to correct the user's input when they misspelled the movie title (note that correcting misspellings and searching for the original titles are features of the search engine of TMDB's API).

[4]Participant P3 did not test Cinemito due to connection problems.

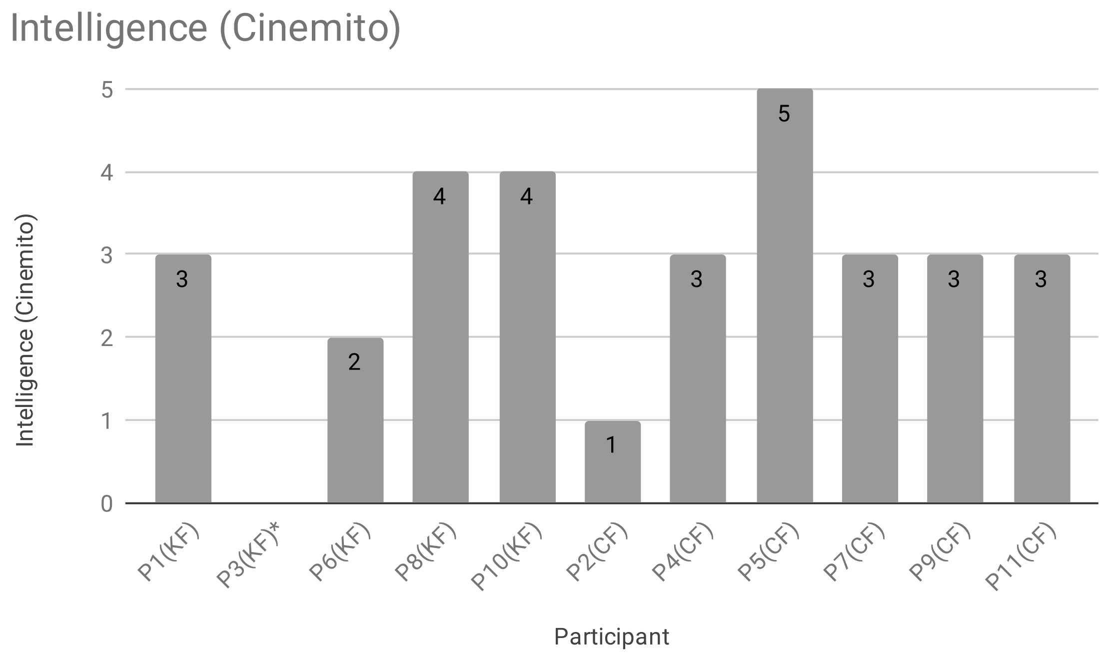

Figure 6.21: Scores for Cinemito's intelligence. Source: Authors

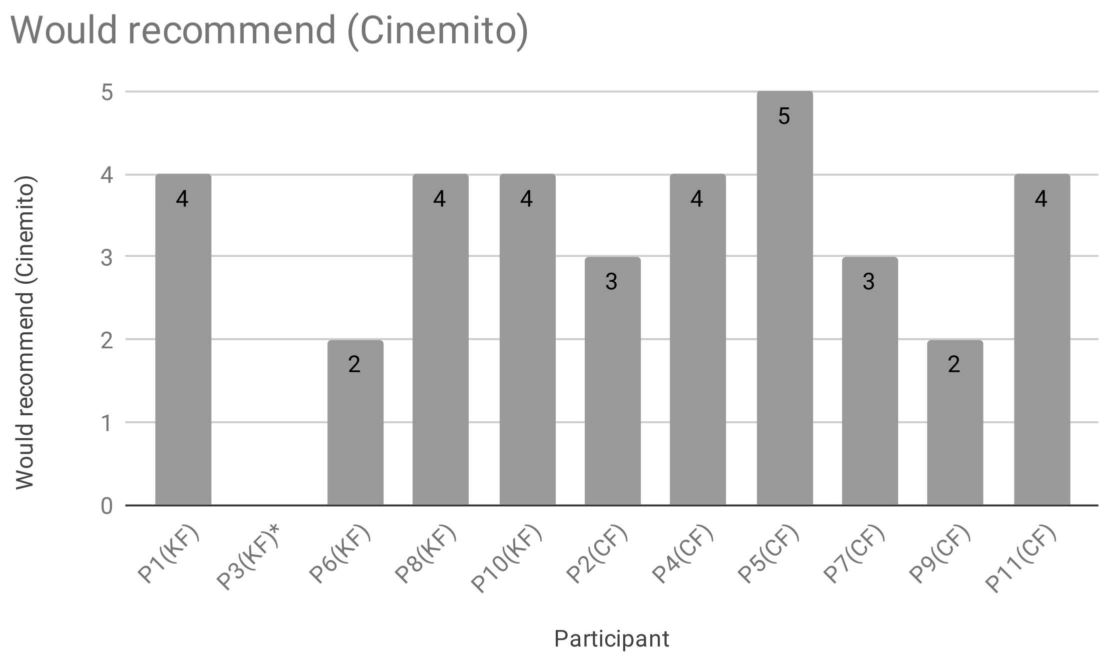

Figure 6.22: Scores for how likely it is to recommend Cinemito to someone. Source: Authors

When asked whether they would **recommend** Cinemito to other people, the mean score was 3.5, the lowest score was 2 and the highest was 5. Figure 6.22 shows these values. The first five columns, marked with KF, show the scores of the participants that tested Kino first[5], while the other columns, marked as CF, contain scores of those who tested Cinemito first. Two participants gave a score of 2: P6 did not want to justify his grading; while P9 said there was not much use for that context (information about movies), and it would be faster to use a browser and just Google the movie title, but P9 also said that maybe using the browser would be slower in a smartphone. Participants P2 and P8 said that searching the movie on Google would be easier, and because of that, it would be hard to convince someone to use Cinemito. P1 and P10 said Cinemito is easy to use, and P11 said it is fun and interesting. Only P5 said Cinemito would be useful in his daily life.

* * *

The next questions were asked only to users that tested Cinemito first, because of that, instead of the ten responses from the previous questions, these were only answered by six people. These questions are related to subsections 2(b) to 2(h) of the interview script and were asked after the users finished testing Cinemito and before they tested the other chatbot.

What **caught the attention** of the users were the posters figures, as noted by P4, P7, and P9, as well as how the *carousel* was used to display the results (P9). P4 liked that the ranking was presented in chronological order (that is a misperception, as Cinemito just uses the same order as the results from TMDB's API, which are ranked by popularity). P5 said the theme of the chatbot is very unique ("It's very different, I had never thought about [using] a chatbot for that"). P7 liked that it is fast and shows the options. Users P1 and P11 said that nothing caught their attention.

As **positive points**, participants P7 and P11 mentioned that Cinemito was easy to use. P4 and P9 also commented about the movie posters that the chatbot uses and the *carousel* when displaying the search results. P7 also mentioned that you do not need to know the commands to use Cinemito. P4 also liked that there is no need to type a lot when using the chatbot. The speed at which Cinemito replied was noted by P2. P5 liked the simplicity of the chatbot, and how it is straightforward to use ("You just have to open it [the chatbot] and type, you don't need to log in or use a menu").

The **negative points** were the following. Participants P4 and P7 said that Cinemito has few features, lacking an option to see the movie's cast or a refined search. P2 and P11 explained that the interaction was not really a conversation and Cinemito could not understand simple things like "hello". Participant P9 considered the size of the poster when a movie is selected as being too big for a smartphone screen. P5 thought that there could be a video tutorial at the start of the conversation (P5 had not noticed the tutorial Cinemito offered in the first message).

[5] Participant P3 could not test Cinemito due to connection problems.

All six participants said to have **enjoyed using** Cinemito, they said that it was a fun and an interesting way to search for that kind of information. Participant P5 also said that Cinemito is simple and straightforward, and having few options helps to keep its focus.

Out of the six people that tested Cinemito first, four (P2, P4, P5, and P11) said they **would use it again**. P7 would only use it if it was a more complete version were made available. P9 said he probably would not use Cinemito again, as he believes it's easier to just make a Google search to get the desired information.

The **suggestions** users gave for improving Cinemito focused on two main themes. The first is increasing the number of features, such as search by cast and actors, showing the actors photos (P4), detecting movie franchises and grouping these movies (P7), learning the user's preferences in order to suggest movies they might like, and sporadically suggesting movies that are playing near the user (if the user authorizes it) or new reviews for a movie (P9). The other theme of suggestions was improving the conversational aspect of Cinemito, like replying to "Hi" and "how are you?" (P11); answering questions as "who won the Oscars?" (P5); and making the conversation more fluid or making clear how the interaction should take place (P2).

Only one participant (P7) made further **comments** about Cinemito, saying its name is adorable, and the system is fast to use and well explained.

6.3.3 Comparison

This subsection analyzes the answers to the questions comparing both chatbots (subsections 3(b) to 3(h) of the script). These questions were made after the users had tested both Kino and Cinemito. As we explained before, there was a problem with Cinemito during one of the tests, because of that, P3 did not test Cinemito and was not asked to compare the two chatbots. That way, these answers include those of the four participants that tested Kino first and the other six that tested Cinemito first.

Table 6.5 shows an overview of the comparative analysis between the chatbots. The second column shows whether that user first tested Kino or Cinemito, while the other columns represent participants' answers to each question.

As seen in the third column of Table 6.5, out of the ten people that tested both Kino and Cinemito, seven (P1, P4, P5, P7, P8, P9, and P10) **enjoyed more** the Cinemito chatbot. Among the three that preferred Kino (P2, P6, and P11), P6 and P11 were not really sure about which chatbot they liked more, P6 even noted that both chatbots are similar, but with different approaches ("I believe they have different approaches to solve

Table 6.5: Comparison results overview

	First test	Preferred chatbot	More complete	More intelli-gent	Easier to use	Understand sentences equally well?	Use NLP?
P1	K	C	C	C	C	K	K
P2	C	K	Equal	K	K	K	K
*P3**	K	–	–	–	–	–	–
P4	C	C	Equal	C	C	C	C
P5	C	C	Equal	K	C	Equal	Both
P6	K	K	Equal	K	C	K	K
P7	C	C	C	C	C	C	Both
P8	K	C	C	C	C	C	K
P9	C	C	Equal	K	C	Not equal	K
P10	K	C	C	Equal	C	K	Both
P11	C	K	Equal	K	C	K	K

the same problem (...) If I was forced to pick one, I would choose the first one [Kino], as it is more versatile"). Amidst those who tested Kino first, three opted for Cinemito, and one preferred Kino. Out of those that used Cinemito first, two liked Kino better (P2 and P11), while the four others (P4, P5, P7, and P9) chose Cinemito.

Those who preferred Cinemito gave different reasons for their choice, the two main reasons were related to an easier interaction (even if more limited) and showing posters of the movies. Most of them mentioned that in Cinemito, users know what they can do next, and there is no ambiguity on what you can do thanks to the use of *quick reply* bubbles (P4, P5, P8, P9, and P10); P9 also said it was more comfortable to use Cinemito due to the *quick replies* ("Although it is dumber [than Kino], I felt more comfortable using this one [Cinemito], it instructs me about what I can do [next]"). P1 and P7 said it is faster and easier to use Cinemito. Most of them valued the use of images of the movie posters, and mentioned that the images help users identifying the movie they want, especially in the case of franchises (P5, P8, P9, and P10); P4 said Cinemito has a better appearance, referring to the use of images and *carousels*. P10 liked that he did not have to type a lot and that the images and the *carousel* help identifying the movie he was looking for. P6, who was in doubt but ended choosing Kino, said that Cinemito seems less versatile (than Kino), but it makes its limitations very clear and, thus, should cause less frustration to users. He also said that it was not clear what the *quick reply* bubbles meant, and he thought they were the auto-complete function of the smartphone's keyboard.

The ones who chose Kino liked that it seems more versatile (P6), and chats more (P11), i.e. answers to small talk questions, and that makes him want to see its limits and try to break it by asking difficult questions. But P11 also said he would probably just use Google to find information about movies instead of a chatbot. P2, who liked the fact that Kino chatted more, said it should use images as well. P4, who preferred Cinemito, complained about the sequence of large blocks of text that the chatbot sends (usually when sending the overview of a movie).

When asked which chatbot **seems more complete**, four people (P1, P7, P8, and P10) said Cinemito was more complete (though P7 and P10 were not sure), and six people (P2, P4, P5, P6, P9, and P11) considered that both chatbots were equivalent in terms of features. None thought Kino was more complete. These answers are shown in column four of Table 6.5 on the preceding page. It is important to note that both chatbots have the same features and use the same database. The four participants that considered Cinemito more complete explained that the reason was that it showed what they could ask it and it seemed to have more options. Three of them (P7, P8, and P10) also said that maybe Kino could also do other things, but they would have to try to find out the commands; P10 said Kino kept him in the dark regarding its features.

Regarding those who said both chatbots had the same features, participant P6 said that Kino creates more expectation, and that leads to frustration when that expectation is not met, while Cinemito has better usability. P4 said that, although both chatbots had the same features, it is harder to find them on Kino. P11 said that each chatbot does something different (regarding the interaction itself), but they equate themselves in the end (regarding their content). Other participants (P2, P5, and P9) just said that both chatbots seemed to present the same content.

About which chatbot **seemed more intelligent**, five participants (P2, P5, P6, P9, and P11) considered Kino as more intelligent, four (P1, P4, P7, and P8) though Cinemito was more intelligent, and only participant P10 considered both equally intelligent. Column five of Table 6.5 on the previous page shows these values. This is an interesting result, as it contradicts the scores given to the chatbots' intelligence in a previous question, in which Kino had a lower average than Cinemito: 2.77 against 3.1. Among those that considered Kino more intelligent, P2, P6, and P9 justified it focusing on the fact that it can answer some questions in natural language. P2 also praised the varying error messages from Kino, that is, it has a series of different error messages for when it cannot understand something. P11 said that Kino tries to emulate human behavior, while Cinemito does not even bother. P5 mentioned that Kino seems to use NLP, but not very well. Participant P10, that considered both chatbots as having equivalent intelligence, explained that Cinemito is more specific on what it can do, while on Kino users have to discover how to do it.

Out of the users that deemed Cinemito as more intelligent, P1 and P4 mentioned the *quick reply* bubbles, saying that the bubbles help users understand what it can do, making the chatbot seem more intelligent. P1 said that Cinemito had a better interface (both chatbots were displayed using Facebook Messenger, so the user is probably referring to elements like images, *quick replies*, and *carousels*), but Kino was able to chat more. Participant P8 said that Cinemito was able to get by on its own (probably referring to fewer errors during use). P7 said that Cinemito seemed more prepared for answering, while on Kino users had to type the sentence the exact way it wanted.

On the topic of which chatbot is **easier to use**, nine users believe that Cinemito is easier to use, while one user considered Kino easier. Column six of Table 6.5 shows these values.

Among users that considered Cinemito as easier to use, six (P1, P4, P5, P6, P9, and P10) cited the *quick replies* as one of the reasons; the *carousel* and poster images were also mentioned by participants P1 and P9. Three users (P7, P8, and P11) said that Cinemito is faster (as direct to the point) than Kino. P4, P7, and P10 explained that Cinemito shows how to use it, by having a tutorial and by using *quick replies*. Participant P10 said that Cinemito ended directing the interaction, always pointing the next possible steps ("It's easier [to use] because of the directions it gives you (...) it even shows the options. But I've ended focusing on what it suggests, maybe it has other things [functionalities]"). P5 said that it was a good thing that there was only one way to ask what you needed, so it's easy to learn. P11 said that, thanks to the images, when you are not sure about the movie title, you still may find it using Cinemito (by flipping through the posters) while you will not find it on Kino. P5 also said that Cinemito always shows *quick replies* of what it can do, so the user knows what it can do, while on Kino, the user does not know what it can or cannot do.

Participant P2, that considered Kino easier to use, said that its error messages were helpful, as well as the fact that you can ask questions directly, making it easier to use.

Participants were asked whether both chatbots could **understand sentences equally well** and **whether one of them did not use NLP**. Their answers are shown in columns seven and eight of Table 6.5 on page 140. It is important to note that, while Kino makes use of NLP techniques, Cinemito does not: it simply uses users' input as a search parameter.

Nine participants considered that the chatbots do not understand sentences equally well. Out of those, five (P1, P2, P6, P10, and P11) deemed Kino to understand better, while three (P4, P7, and P8) believed that Cinemito was able to better understand the sentences. Participant P9 considered that, although the chatbots did not understand sentences equally well, they could not be compared, as they took different approaches to their designs: "The first one [Cinemito] seems to work using keywords, but it does not understand if I add something else [to the keyword], because I tried 'Indiana Jones direction' and it did not work. And this one [Kino] wants the question 'what is the director...' (...) So they work in different ways, different approaches". This user also thought that Kino made use of some form of NLP, but it was not enough, as its understanding was still "quite weak". He also considered that NLP is an important feature for a chatbot, as it creates value.

Participant P5, who considered that both chatbots understood sentences equally well, also considered that both made use of NLP techniques. When asked to clarify, P5 said that both chatbots seemed to make some kind of processing of user's inputs. "The first one [Cinemito] (...) when I type something, let's say 'Harry Potter', it shows a list of movies then I choose the one I want (...) I would choose the first movie and press 'direction'. (...) Here [on Kino] it's more complicated because of the long title. I typed 'Who directed Harry Potter' and it returned the 'Harry Potter and the Order of the Phoenix', but I want 'Harry Potter and the Chamber of Secrets', so I had to type the question again with the exact title [to get the information]".

P4, one of the users that considered Cinemito to understand sentences better, said that the use of *quick replies* allows Cinemito to better understand the user, as on Kino the user does not know what to say to the chatbot because the same thing can be said in many different ways, and the chatbot cannot understand them all ("I think the first one [Cinemito], as it gives you the options to ask, it understands better, but that is because it gives you the options to talk to it. As the second one [Kino] does not give options to talk to it, it is harder (...) It is vast... You can say anything: it is saying 'What is the story of the movie Harry Potter' [showing the message she had sent on the screen], you could say 'overview', 'synopsis', or 'tell me about it', or anything like that. So, as the first one gives me the options, it is easier for it to understand me, but that is because it made itself understood (...) We can say anything in many ways, but it [the chatbot] is not human, it will not understand everything, so if it explains how it can understand, it is easier"). This participant also said that the way you have to form the sentences for Kino to understand sounds a little off, some of them do not sound very "natural", as in the way people usually speak.

Out of the other two participants (P7 and P8) that answered that Cinemito could understand the sentences better, P8 considered that Kino was the only one to use NLP techniques. These users considered NLP important, but, in this case, Cinemito proved to be useful without using it. P8 also explained that more advanced NLP could be handy for complex queries, such as "what is the age of the movie's director?".

Five participants (P1, P2, P6, P10, and P11) considered Kino to understand sentences better. P10 believed that both chatbots use NLP, while the others deemed that Kino was the only one to use it. P6 said Kino seems to use some kind of pattern matching, while Cinemito only looks up in a database for answers. P1 was indifferent regarding NLP in chatbots, while the others (P2, P6, P10, P11) considered it important. Out of those, P6 said that it is important that the chatbot make its limitations clear (as in informing which kind of sentences it can understand and providing examples). Participant P11 said that the conversation is what makes him use a chatbot, otherwise, he would just search on Google ("It makes no sense to have a chatbot and not use natural language"). P10 said that NLP is important, but a chatbot can be helpful even without it by guiding the conversation or by using very specific commands, akin to a command-line shell, depending on the use of the chatbot.

Regarding their **previous experience with chatbots**, all of the participants had previous interactions with chatbots, especially personal assistants, like Siri and Google Assistant (P4, P5, P8, and P9 mentioned them). P1 and P2 cited a dictionary chatbot that was available on Google Talk, in which you type a word and the chatbot replies its definition. Participant P6 mentioned Bradesco's (a Brazilian bank) Facebook Messenger chatbot. P7 had used a chatbot for searching Magic The Gathering (a card game) cards on Telegram. Two users (P10 and P11) had already chatted with Turing-test-focused chatbots, such as ELIZA, which can talk about any subject.

While **comparing previous experiences** against Kino and Cinemito, among participants that had experience with assistants like Siri and Google Assistant, P5 said that Siri has a personality and feels more humane than Kino and Cinemito, although P5 noted that these also try to simulate a person ("It is different, due to its personality [Siri's] (...) Siri sounds more like a person, but these [Kino and Cinemito] also simulate a person"). P9 said that you can talk to Siri, and that is better than typing, P9 also noted that, in general, Siri is closer to Cinemito than to Kino, but with more processing power. Participants P4 and P8 noted that the scope of the chatbots is more limited than those of personal assistants: P4 said that Google Assistant has many more options, but she did not like that it frequently replies answers with links to outside of the chat environment; P8 said Kino and Cinemito have more restricted domains, and that leads to fewer errors, for as the chatbot has a smaller domain, it is more predictable and users know what it can talk about, so they will not ask something out of scope, while in the case of Siri and Google Assistant, that have a broader scope, they get some questions right, but still make a lot of mistakes.

The two participants (P1 and P2) that had previously used the dictionary chatbot on Google Talk said that it was practical and direct to use, just like Cinemito. Participant P6, that had used the Bradesco chatbot, said that its quality was not very different than that of Kino and Cinemito's, and he believes that chatbot technology as a whole is not good yet. Participant P7, that used a chatbot for looking up Magic The Gathering cards on Telegram, said it was more restrictive than Kino and Cinemito, as it expected the user to know its commands in advance to use it, while Kino and Cinemito are more comprehensive and simulate a conversation, instead of just being a query to a database. Finally, P10 and P11, which had used ELIZA and other conversational chatbots, understood that these chatbots have different purposes than Kino and Cinemito: ELIZA and other similar chatbots can understand the sentences better and keep a conversation, while Kino and Cinemito cannot do it, but there is no need for them to do it, as their context is more focused.

6.3.4 Strategies and Sign Classes

In this subsection, we analyze what the participants answered to the questions in section 4 of the script, regarding some of the strategies for informing users about features and a few of the sign classes, as well as about a feedback feature present on Kino, to check whether they had noticed these elements while interacting with either of the chatbots and their opinions about those elements. These questions were asked after users had already tested both chatbots and answered the previous questions. We asked these to a total of ten people, as P3 did not interact with both chatbots.

Table 6.6 shows these results. The first two columns show the participant and whether they used Kino or Cinemito in their first test. The following columns represent participants' answers to the questions in section 4 of the script.

Table 6.6: Strategies results overview

	First test	Message informing features	Tutorial	*Persistent menu*	Help option	*Quick replies*	*Carousel*	Feedback
P1	K	None	None	None	None	C	C	None
P2	C	K	None	None	None	C	C	None
*P3**	K	–	–	–	–	–	–	–
P4	C	Both	C	C	None	C	C	None
P5	C	C	None	None	K	C	C	None
P6	K	None	C	None	None	C	C	None
P7	C	C	C	None	None	C	C	None
P8	K	C	C	None	None	C	C	None
P9	C	Both	C	None	K	C	C	None
P10	K	Both	C	None	None	C	C	None
P11	C	Both	C	C	K	C	C	None

We asked users if they had noticed any **messages informing what the chatbot could do**, in reference to **S1** (*Showing the main feature on the first message*), **S8** (*Showing the main menu or the most frequent features when user asks for help*), and **S9** (*Showing the main menu or the most frequent features when user asks for help*). The third column of Table 6.6 shows participants' answers. Both chatbots have this kind of messages implemented: Cinemito informs users about what it can do during the first messages (**S1**) and when users select "help" in the *persistent menu* (**S8**); Kino [6], likewise, also informs what it can do when users ask for help (**S8**), and when they type something the chatbot cannot understand (**S9**).

[6]Kino also implements **S1** (*Showing the main feature on the first message*), but an error prevented its first messages from being shown to participants during tests.

Although all participants came across these types of messages in their interactions with both chatbots (either on Cinemito's first message or when Kino did not understand what they typed), when asked about it, not all users recalled them. Eight participants said they had noticed these messages. Out of those, four (P4, P9, P10, and P11) noticed it on both chatbots, three (P5, P7, and P8) noticed it only on Cinemito, and one participant (P2) noticed it just on Kino. Participants P1 and P6 have not noticed these messages in any of the two chatbots.

When asked what they thought about such messages, participants P1, P4, P9, P10, and P11 said it would be very important at the start of the conversation, also explaining that it would make it easier for users if the system made clear what to expect, avoiding frustrations. Three participants (P4, P6, and P10) also said that such messages would be useful when errors happened. One user (P11) mentioned Cinemito's tutorial as an example of a message informing that the chatbot can do, he also considered that a message informing what the chatbot can do when users type "help" would also be useful. Participant P7 said such message should at least inform which commands the chatbot can understand.

Note that, as we stated before, due to a problem with Kino, its first message informing what it can do was not shown to users during tests. Even so, participants were able to notice similar content on some of Kino's error and help messages.

The question regarding whether participants **noticed a tutorial** refers to strategy **S2** (*Guiding the user through a short tutorial during first messages*), that was only implemented in Cinemito, whose first messages offer the tutorial to all users. Participants' answers can be seen in column four of Table 6.6. Out of the ten participants, seven (P4, P6, P7, P8, P9, P10, and P11) noticed it when using Cinemito (P9 was not very sure if he had seen it or not); while three others did not notice it in any of the tested chatbots.

Note that the way that Cinemito's tutorial is implemented allows users to opt out of it during any moment of its duration, nevertheless, as we have said before, Cinemito chatbot offered the tutorial to all participants on its first message. During tests, only participants P4 and P11 followed the whole tutorial until the last message. P8 and P10 accepted the tutorial offer, but did not follow it completely, opting out of it when the chatbot suggested searching for "The Matrix". P5, P6, and P7 selected the tutorial option when offered, but ignored it completely, maybe they did not fully read Cinemito's sentence regarding the tutorial (when asked about the tutorial, P5 did not recollect any of the chatbots offering it). Participants P1, P2, and P9 declined the tutorial when the chatbot offered it.

We also asked participants' opinions regarding the use of a tutorial on a chatbot. Five participants (P4, P5, P7, P10, and P11) considered it important, as it would help to inform users about how to interact with the chatbot and what it can do (P4, P7, and P11). Four participants (P4, P9, P10, and P11) said it should be shown on the first message, out of those, P9 and P11 were very emphatic that it should be shown just during the first time using the chatbot and, after that, just an error message saying how to use the chatbot would be enough. Participant P4 also suggested that the tutorial could be also offered during error messages. P2 said a tutorial should be short, while P5 said it could be a video tutorial, both participants had not followed nor noticed Cinemito's tutorial. Four participants (P1, P6, P8, and P9) said they would not follow a tutorial. out of those, P6 and P8 also said that users should learn as they go and that a good error message should be sufficient to guide them (as P6 puts it: "*It is better to have good error messages than a tutorial*"). Finally, P6 liked that it was possible to opt out of the tutorial.

Regarding whether they noticed the **persistent menu**, related to **S4** (*Having a persistent menu with main features*), column five of Table 6.6 shows the participants answers. Note that both Kino and Cinemito have *persistent menus* implemented. In both cases, to access the *persistent menu*, users have to click the *hamburger button* on top of the keyboard, at the right of the input bar and the smiley emoji. Figure 6.23 depicts a screen capture of Cinemito, showing both the system's keyboard and, on top of that, the input bar and the menu button on the right.

Only participants P4 and P11 said they had noticed it during the tests, both on Cinemito, while the other eight participants said they did not see any menu during the tests. But P4 and P11 mistook the *quick reply* bubbles for the *persistent menu*. During his test with Kino, P11 stumbled upon its *persistent menu*, but did not consider (or did not recollect) it as a menu when asked this question, instead answering that the *quick replies* were the menu. Participants P2 and P5 also interacted with Kino's menu during their tests but answered that none of the chatbots had a menu.

In general, the participants did not like the idea of a menu on a chatbot: six of them (P1, P4, P6, P7, P8, and P9) said they would not like it, as it would clutter the interface; P9 added that a menu is "conceptually bad" for a chatbot, not fitting into a conversation. P7 said that it could be useful, but it should be very discreet and stay hidden most of the time. Participant P8 said that a chatbot should be simple enough so you do not have any need for a menu. P6 suggested that the menu could be made within the chat interface by using a *carousel*, instead of outside of the chat area (note that the *persistent menu* is not in the chat area).

On the other hand, participant P11 (one of those that stumbled upon the menu during tests), after being informed that *that* was the *persistent menu*, said it was hard to find, and it was not clear that it was a menu, but it is an important feature.

It is important to note that, as we said before, the tests were conducted using the mobile version of Facebook Messenger. Being on the mobile version, the *persistent menu* replaces the text-input area. Even so, it was hard for users to notice it as a menu. That may have happened because most of the participants had never used a chatbot on Facebook Messenger before. During tests, participant P6 opened Facebook Messenger's

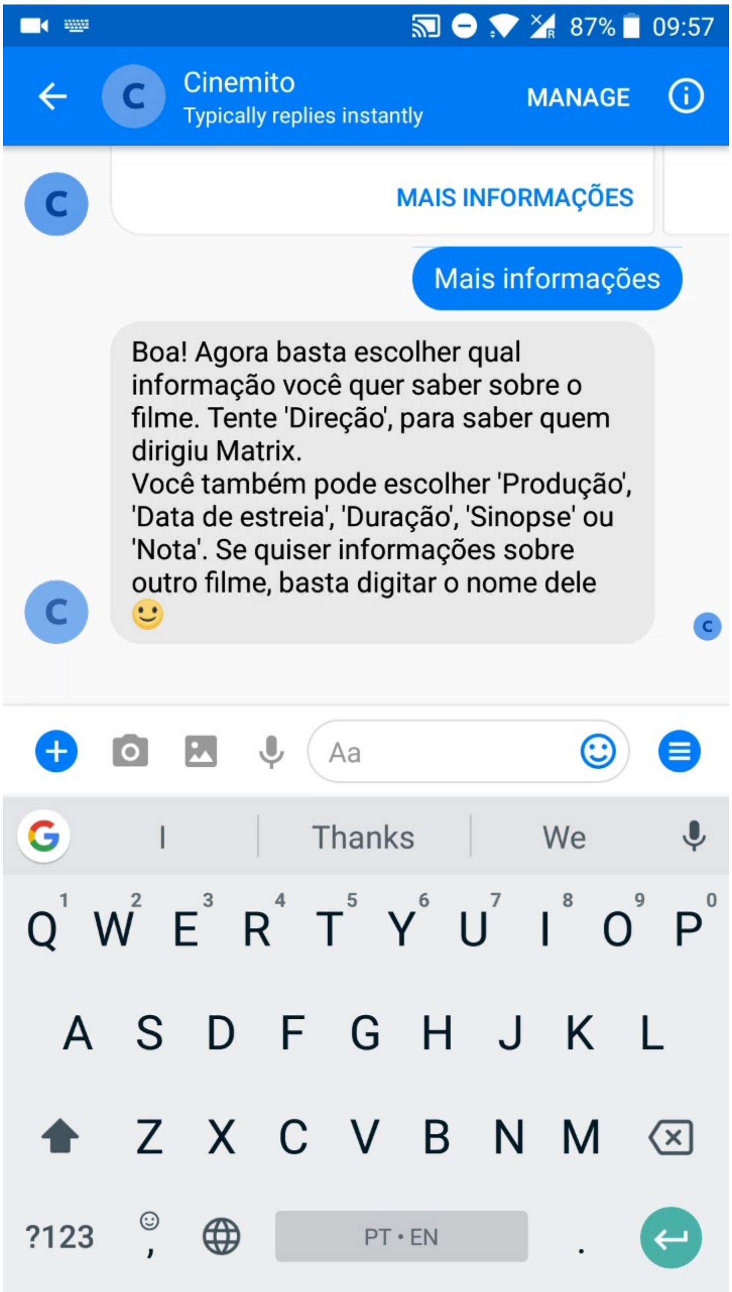

Figure 6.23: Cinemito showing the system's keyboard. Source: Authors
Translation: *–More information*
–Great! Now you just have to choose which information you want to know about the movie. Try "Direction", to know who directed The Matrix. You can also choose "Production", "Release date", "Running time", "Overview", or "Rating". If you want information about another movie, just type its title :)

menu thinking it was the chatbot's *persistent menu*.

That way we have evidence pointing that, besides being hidden and easy to miss, the way the *persistent menu* is shown in Facebook Messenger on smartphones is also not obvious that it is a menu indeed, as highlighted by the users that have interacted with it but did not consider it a menu.

Concerning whether the participants found the **help option** present in the chatbots, the answers can be found in column six of Table 6.6 on page 145. Both chatbots have a help option: Cinemito has an option under its *persistent menu*, while Kino displayed a help message when users typed "*ajuda*" (Portuguese for "help") besides also having an option for it in its menu.

Only three participants (P5, P9, and P11) answered they saw the help option on Kino, P9 also said that maybe there was an option like that on Cinemito. The other seven participants said they had not found any traces of a help option in any of the chatbots.

When we asked their opinions about a help option, the participants who found it on Kino (P5, P9, and P11) said that it is important, because a user may not remember what the chatbot can do and how to do it. But P5 and P11 also complained that Kino's help messages were not good, as they did not give much detail about its features nor on how to access them, as P5 said: "It is the same [error] message over and over, but it does not detail how to do it". Participant P9, who said that maybe Cinemito had a help option as well, suggested that the *quick reply* bubbles can function as help to users.

Out of the seven participants who said to have not found help options in any of the chatbots, P1, P4, P7, and P10 said such an option is important, while P2, P6, and P8 said it was not necessary. Participants P4 and P10, who deemed a help option important, said it should be shown at the beginning of the conversation or after errors, P10 added that the help option should only be shown after a few repeated errors, as that would indicate the user is having difficulties. Participants P6 and P7 said they would look for a help option in the menu. Participant P10 also said that he did not feel much need of such an option on Cinemito, but it was necessary on Kino. Another participant (P6) said that, although he thinks a help option is important, he would not use it, preferring trial and error for discovering how to use the chatbot.

Participant P2, who said the help option was not necessary on a chatbot, said that just the initial message explaining the features is enough. P8 claimed that the chatbot needs either the instructions on how to use it or the help option, having both is doubting the user's intelligence. P8 also added that a help option could be confusing for the user (thinking about a dedicated help button on the interface): "If it were my grandfather using it [the chatbot], he would be confused with it [the help option]".

About the **suggestions of replies** (*quick reply* bubbles), related to **S3** (*Suggesting the next possible set of actions to the user*), column seven in Table 6.6 on page 145 shows users answers: all of the participants noticed them only in the Cinemito chatbot (Kino does not make use of *quick replies*, after all). However, participant P7 was not sure whether Kino also had suggestions of replies or not, as he considered that the error messages could be counted as a suggestion of reply.

The comments about the *quick replies* were the following. Eight participants (P1, P2, P4, P5, P6, P8, P10, and P11) considered it a positive feature, saying it makes the chatbot easier to use (P5 and P9) and saves time as the user does not have to type (P2, P5, P9, and P10). Some participants considered that *quick replies* also serve as a way for the chatbot to show what it can do (P4, P8, P9, and P10), and that it contributes to its predictability, P10 said he could be certain that by selecting the *quick reply*, the chatbot would provide information about the topic in the label, without errors ("I am sure that it [Cinemito] will return me the information I select [in the *quick reply*]"). Participants P4 and P9 also said that Kino could benefit from the *quick replies*, so users would not have to type so much. P11 also considered that Kino could use *quick replies* to help in cases of disambiguation of different movies with similar titles for example (very similar to the use *carousels* have on Cinemito).

As complaints regarding *quick replies*, participants P2 and P9 said that they prefer a conversation with the chatbot, and the *quick replies* are not really like a conversation. P9 also said that the *quick replies* would not be necessary if the chatbot could really understand natural language. Nonetheless, P9 concluded that the *quick replies* help to guide users on what the chatbot can do and, although it departs from the conversation metaphor, it still helps, similar to a person showing a list to another. Participant P7 said that *quick replies* can be useful when dealing with fewer options, but they may become unfeasible when there are many alternatives for the user to choose from. Participant P6 complained that the *quick reply* bubbles should differentiate themselves more from the keyboard's auto-complete function.

When we asked users whether they noticed the **carousel** of movie posters, all ten of them said they saw it while using Cinemito. That was expected, as Kino does not make use of the *carousel* sign class. Their answers are summarized in the second to last column of Table 6.6 on page 145.

About the usage of the *carousel* in Cinemito, seven participants (P1, P4, P5, P6, P8, P9, and P11) considered it a good feature for the chatbot. P2, P9, and P10 liked that it showed the movie posters and that these images were not too large (P9). Participants P10 and P11 said that the *carousel* with the movie posters helped when searching movies within a franchise, as they could search the franchise name and then select the desired installment. Participants P2 and P5 said that the posters in the *carousel* helped to select the right movie among others with the same or similar name. P1 and P4 said that using the *carousel* is intuitive. P4 also liked that, by displaying the *carousel* with posters, the chatbot shows that it also has content about other movies. Participant P8 complained that the order the movies are shown in the *carousel* is not very good, and the chatbot should inform the criteria used for ranking them.

Finally, we also asked users about the **feedback feature**. The last column of Table 6.6 on page 145 contains users' answers about whether they found such feature in any of the chatbots. The feedback feature is only available on Kino: when the user says something like "I like it", "thank you", or "the answer is wrong" Kino considers it as feedback to its previous answer. During tests, none of the participants stumbled upon that feature. Consequently, none of the users noticed this feature in any of the tested chatbots.

When we asked their opinion about a feedback option in a chatbot (without showing them the feature on Kino) participants P2, P7, and P10 said that it is a good idea, especially for the chatbot's developer. P4 and P9 said that it might be a good feature, but the chatbot should explain what would be done with the feedback. Participants P1, P2, P6, and P8 said that they would not use such feature, as it would be bothersome. P6 said that, if he decided to leave feedback, he would do it on App Store or Google Play Store. Participants P5 and P6 considered that the feedback option should be separated from the chat environment, so it would not be confusing. P9 and P11 said that the feedback option should be inside the chat, as part of the conversation, maybe with the user saying "I would like to leave my feedback".

After being informed that indeed there was a feedback feature on Kino (but not knowing how to access it), participants P6 and P11 asked to look for it. After a while, they both found the Facebook Messenger option for leaving feedback *about* the chatbot to the Facebook Messenger team, instead of finding the intended feedback option within the conversation with Kino.

6.3.5 Considerations

This subsection shows the main insights that emerged from the analysis of the user tests with both chatbots. It is important to emphasize that both Kino and Cinemito were prototypes when tested, that way some of the points that caused frustration in participants can be derived from that. However, as our focus in this work is the chatbot's way of communication, we have interesting results that are detailed below.

By analyzing the answers for **Kino**, we notice that participants like the idea of chatting with the system and consider a chatbot that can chat more interesting than one that can not, as it would be more akin to a query to a database. But we could also notice that the large communication space of possible messages users can send to the chatbot, associated with a restricted set of type of sentences that it can understand, may have led

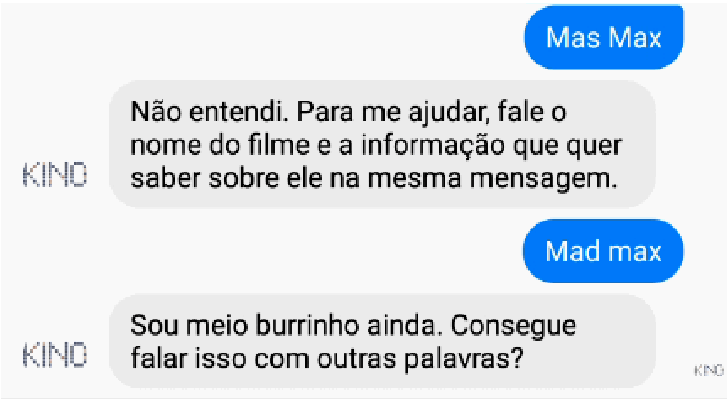

Figure 6.24: Kino's error messages. Source: Authors
Translation: –*Mas Max* [sic]
-I did not understand. To help me, say the movie title and the information you want to know about it in the same message.
–Mad Max
–I am still a little dumb. Can you say that in other words?

to difficulties in the interaction and communication breakdowns. This is aggravated by only a few messages being available to inform users how to use the chatbot. In the case of our test, the fact that Kino's initial message was not presented, might have increased the difficulty perceived to interact with it and/or might have had a negative impact on their experience with the chatbot as a whole.

The fact that Kino did not keep the conversation context between messages (a problem common to both tested chatbots) also caused frustration to users. In this case, if the participant wanted to know the name of the producers of the movie they had just asked about, they would have to type the full sentence again, including the movie title. Another interesting fact is that a participant mentioned the confusing way the information is presented, indicating that the chatbot violated the maxim of Manner from the Cooperative Principle (avoiding obscurity and ambiguity, and being brief and orderly, see Chapter 5 - "Pragmatics and Chatbots" for more information). That frustration was also noticed when participants were asked whether they would recommend Kino to someone.

Nevertheless, Kino's friendliness and politeness were praised. Especially during error messages, in which the chatbot "took the blame" for the error, as seen in the example of Figure 6.24, in which the chatbot says that it is still a little dumb, implying it is its fault for the communication breakdown, which is a good example of the maxims of Modesty (giving low value to S's qualities, self-devaluation) and Obligation of S to O (giving high value to S's obligation to O, case of apologies and thanks) from the Politeness Maxims (again, see Chapter 5 - "Pragmatics and Chatbots" for more information).

When analyzing what was told about **Cinemito**, although the lack of conversation was pointed, with participant P2 saying it is a “chatbot that does not chat”, Cinemito’s scores for ease of use and intelligence were higher than Kino’s (though when asked which chatbot was more intelligent, Kino had one extra vote). That perception occurred probably because Cinemito presents what it can do, and its rules are made clear to the user through the use of tutorial, *quick replies*, and messages explaining its features (as some participants pointed when asked which chatbot they preferred). That way, users already knew what to expect from Cinemito early in the interaction, reducing frustration. These strategies helped to close in the communicative space between user and chatbot at the cost of making the interaction less like a conversation. These factors may also help explain why Cinemito had higher scores regarding its Intelligence: as users encounter fewer breakdowns during the interaction, their perception of the chatbot intelligence can increase, and the other way around can also be true, as when many errors occur, users may tend to consider the chatbot as less intelligent.

Another situation that happened was some participants not fully reading messages the chatbot sent. That was observed when users replied “yes” to Cinemito’s tutorial, and then opted out in the next interaction, and afterward mentioned not having noticed the tutorial. This could be an effect of the test itself, and (some) participants (not) being eager to finish the tasks. That type of behavior (ignoring or not carefully reading explanations about how to use and error messages) happened during tests of both chatbots. That may be indicative that these messages were not crafted in a way to catch the user’s attention. Perhaps a simpler message showing just the template of what the user can do would work better. A detailed analysis of the causes, as well as ways to improve messages, would be interesting as further work.

During the interview, it was clear that some participants had not noticed that Kino and Cinemito had the same features. The latter showed its features in the tutorial, in the help messages, and, mainly, through its *quick reply* bubbles. That way, participants may have (correctly) inferred that the *quick replies* represented the topics Cinemito could talk about. As participant P10 said, he was sure that if he clicked on one of the *quick reply* buttons, the chatbot would return that information. On the other hand, while interacting with Kino, participants were not clearly presented with what the chatbot could do, or how to ask it. The information had to be explicitly accessed through “help” and “about” options. Furthermore, Kino’s error messages did not inform all of the available features. That way, if the participant had not paid attention to, or had not triggered, the messages presenting Kino’s functions, they would have no alternative other than trial and error to discover Kino’s features. Nevertheless, every user knew at least a single feature of Kino’s: that it can inform the director of a movie, as it was asked in the scenario of the tests.

Another problem that may have made interaction with Kino harder for some users is that its presentation, help, and “what I can do” messages inform the user about *what* it can do, but not *how* to do it. The only message that gives a hint about how to use the

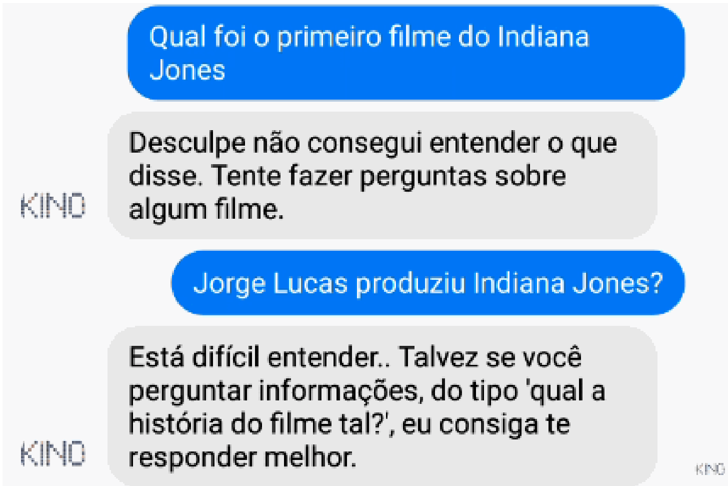

Figure 6.25: Kino's other error messages. Source: Authors
Translation: –*Which was the first Indiana Jones movie*
–*Sorry I could not understand what you said. Try making questions about a movie.*
–*Did Jorge Lucas* [sic] *produce Indiana Jones?*
–It is hard to understand.. Maybe if you ask for information, like 'what is the story of such movie?', I can better answer you.

chatbot is one of its error messages, which are cycled through when it cannot understand the user's input. That message can be seen in Figure 6.25. It gives a valid example ("what is the story of such movie?"), but does not explain that the required order of the sentence is first the topic and then the movie title. That case could be considered as a violation of the maxim of Quantity (of information to be provided, that should neither more nor less informative than necessary) of the Cooperative Principle, as the chatbot gives less information than necessary (only *what*, but not *how*). Furthermore, other error message says the *opposite*, informing that users should "say the movie title and the information you want to know about it on the same message", as seen in Figure 6.24 on page 152, which could confuse the user or lead them to write sentences that Kino will not understand. This last example seen to be a case of violation of the maxim of Quality ("Try to make your contribution one that is true") of the Cooperative Principle.

One other possible way to inform users about the chatbot's features would be to use the *persistent menu*. Although both Kino and Cinemito did not show their features in their menus, the options for help and "what can I do" were there, so users could select it to see a metalinguistic message telling them how to use the chatbot. But not many participants noticed the *persistent menu* during the tests. Even though they were using the smartphone app version of Facebook Messenger, in which the menu is shown instead of the text-input box. Users just clicked on the "Send a message" option and never looked back at the menu. That, of course, may have been the case on our tests because almost all of the participants (10 out of 11) mentioned not having previous experience with chatbots

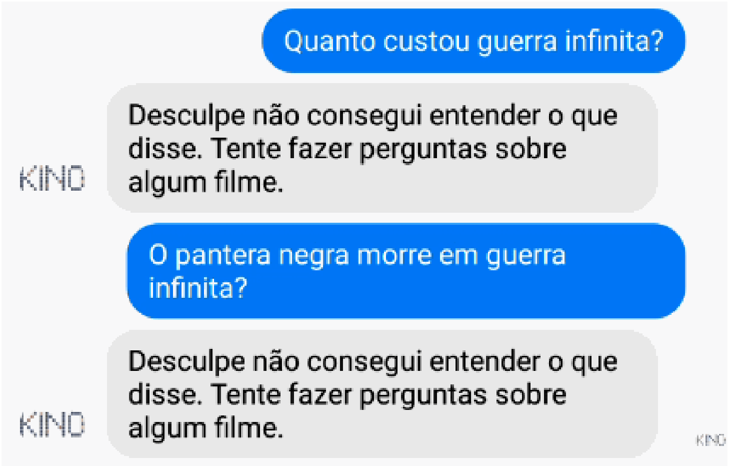

Figure 6.26: User asking Kino a complex question. Source: Authors
Translation: *–How much did infinity war cost?*
–Sorry I could not understand what you said. Try making questions about a movie.
–Does black panther die in infinity war?
–Sorry I could not understand what you said. Try making questions about a movie.

on Facebook Messenger and, thus, no previous knowledge of the depiction of the menu in the interface. Even so, that is a point to be further explored in other works.

When using natural language on the chatbot, some users tried to ask more complex questions to see at which point Kino would "break". An example of that can be seen in Figure 6.26, which depicts some of the questions made by P10. Besides being considered fun by some users, it can also be seen as the users trying to understand the limits of the conversational space of the chatbot. Starting with simple questions, users iteratively try to expand the type and complexity of questions they can ask the chatbot and still be given a valid answer. When they reach the point in which the chatbot cannot understand anymore, they know what kind of questions and complexity they can ask it. But again, this happened during an evaluation of the chatbots, and we cannot affirm that would happen during real use context. Besides that, not all users tried it, and our participants were mostly from the STEM field, so designers should not rely on this kind of behavior when teaching users how to use their systems. It is probably better to make use of metalinguistic signs to inform users how to use the chatbot.

When comparing Kino and Cinemito to intelligent assistants like Siri and Google Assistant, some users explained that these assistants had a broader scope, which could lead to many more errors. That line of thought can be linked to the consideration of the immense interaction space that conversational interfaces have. As participant P8 said: when a chatbot has a narrower domain, a theme, users will focus their questions on the scope of the system. It makes no sense to ask about stocks to a weather chatbot.

The use of *quick replies* and *carousel cards* was compared to a person showing a list during a conversation. To some participants, their use was also linked to the perception of intelligence of the chatbot, by helping users understand what it could do. But mainly, *quick replies*, *carousels*, and *cards* were also related to what users perceived as making the chatbot easy to use. These signs show users what the chatbot can do and also serve as shortcuts, saving users from typing.

Even so, both chatbots would probably be replaced in favor of a simple Google search by the users. But that can also be influenced by their simple purpose and few features, as they are not final products.

6.4 Conclusions

In this chapter, we continued our research through the analysis of user tests and semi-structured interviews. Using the results of previous chapters, mainly the strategies for conveying what a chatbot can do and the sign classes identified in Chapter 4 - "Communicative Strategies Investigation", we decided to analyze users' reactions and opinions about different ways of interacting with chatbots.

We contacted a fellow MSc student who had created a chatbot called Kino that relies on NLP and had a very conversational approach, using only text to communicate with users. We then created another chatbot with the same functions but with another approach to its design and interaction, heavily relying on conversational aids, such as *quick replies* and *cards* on a *carousel* for showing users their options. We also used some of the strategies from Chapter 4 on the design of this new chatbot, which was named Cinemito.

With these two chatbots, we proceeded to design tests and semi-structured interviews with users. We developed a scenario that would make users navigate through some of the features of both chatbots, requested them to think aloud, and asked them about their opinions about the chatbots, how these chatbots compared to each other, and their previous experiences with chatbots. In the end, we also asked them explicitly about if and how they perceived some of the strategies encoded in the chatbots.

This work is a qualitative study with few participants, to allow for an in-depth analysis. So a limitation that stems from it is that we recruited people from STEM fields with programming experience and very familiar with technology in general. So their points of view may not represent that of an average user. On the other hand, by recruiting specialists with a better understanding of the underlying technology, they would raise topics the average user would not, as illustrated by the participants trying to "break" the chatbots.

The analysis of the interviews revealed that there is not a clearly preferred way of interaction. Some users enjoyed talking to Kino and liked its ability to engage in small talk answering messages like “Hi”, or “Hello”, but they also were underwhelmed by its inability to understand variations of the same question. On the other hand, some users also liked Cinemito, saying it was easier to use and faster to get the information they need, but they also missed the capacity to reply to small talk and felt that talking to Cinemito did not feel like a conversation.

We also noticed that some of our conclusions from Chapter 4 - “Communicative Strategies Investigation” found echoes in some of the interviews, mainly regarding the openness of the interaction space of conversational interfaces and the dilemma of limiting that space with aids (like *quick replies*, *carousels*, and menus) at the cost of making the interaction feel more like navigating a menu instead of a conversation.

Another topic that should be highlighted is what users consider as intelligence in a chatbot. In our tests, as the participants were specialists, we could suppose that they would consider the use of Artificial Intelligence, as many did indeed, referring to the language processing techniques and pattern matching. However, it was unexpected to notice some of them relating communicability to intelligence. Some participants considered that one chatbot was more intelligent than the other for better presenting its communicative and interactive capacities to the user. That way, different factors may impact what specialists consider as intelligence, and we may assume that the same may happen to the average user. Thus it would be interesting to further investigate how that factor, as well as other ones linked to interaction quality, can affect the perception of intelligence of a chatbot.

An interesting future work is to use the recording of the interviews for testing the applicability of the Communicability Evaluation Method on chatbots.

In the next chapter we conclude this work by showing our contributions, and future works.

Chapter 7

Final Remarks

In this final chapter, we present the main contributions and limitations of this work. It also contains possible future works to complement it.

A new wave of interest in chatbots and conversational interfaces is happening. These types of interfaces are becoming more common every day. But there are still not many works for supporting chatbots' designers on important design decisions. There are also few qualitative methods for analyzing these chatbots. This work is a step in that direction, as it focuses on strategies for conveying chatbot's features to users while showing a methodology that can be used for generating new knowledge about this domain.

The main contributions of this work are the following:

- **Identification and consolidation of six sign classes** used to compose the chatbots' interface. In our analysis of three chatbots and posterior consolidation with the other ten chatbots presented in Chapter 4 - "Communicative Strategies Investigation", we identified six sign classes:

  - *Simple message*
  - *Simple image*
  - *Quick reply*
  - *Card*
  - *Carousel*
  - *Persistent menu*

  Designers who are developing chatbots could benefit from considering them when conveying information to users. During further tests with users in Chapter 6 - "User tests", some participants praised the use of some of these sign classes as they made it easier for them to discover the chatbot's features. Our analysis and consolidation focused mainly on chatbots using Facebook Messenger as the platform, so although the sign classes identified are generic, other platforms (such as Telegram or Skype) may have different representations for these sign classes or even different ones. Thus, a similar investigation on other platforms available for developing chatbots would be interesting to further consolidate these classes and verify if other considerations are necessary for a broader approach.

- **Identification twelve strategies for conveying chatbots' features to users** and further consolidation of eleven of those (evidences of the 12$^{\text{th}}$ strategy were found during consolidation of the first eleven). Through the use of the Semiotic Inspection Method on three selected chatbots we were able to identify some strategies used by their designers to make their users know what they can do. The eleven strategies are the following:

  - **S1** (*Showing the main feature on the first message*)
  - **S2** (*Guiding the user through a short tutorial during first messages*)
  - **S3** (*Suggesting the next possible set of actions to the user*)
  - **S4** (*Having a persistent menu with main features*)
  - **S5** (*Having a main menu with main features*)
  - **S6** (*Having a list of available commands*)
  - **S7** (*Offering contextual help about a feature*)
  - **S8** (*Showing the main menu or the most frequent features when user asks for help*)
  - **S9** (*Showing the main menu or the most frequent features when user asks for help*)
  - **S10** (*Showing the persistent menu instead of a text-input box*)
  - **S11** (*Highlighting the most important features*)

  Later these strategies were consolidated through a top-down approach inspecting another ten chatbots while looking for evidence of the use of the strategies. During the consolidation, we also found evidence of the 12$^{\text{th}}$ strategy: *Actively offering features to users*. Some of the strategies helped design a chatbot prototype for user tests and, in general, users had a good perception of them. That way, these strategies can be useful for helping designers of chatbots make informed decisions when designing their systems. Possible future works include consolidating the 12$^{\text{th}}$ strategy found during inspections. As the inspection that found the strategy was not triangulated, new inspections using SIM or other compatible methods can be used for shedding light on that.

- **Discussion of approaches for dealing with the openness of the conversational space** can also be of help for designers when considering which approach to take concerning this peculiarity of conversational interfaces. We discussed the large conversational space that chatbots may entail, which can lead to users not knowing what they can say to it. Some of the strategies and sign classes can be used to narrow down that conversational space, but it comes at a cost of limiting how users can express themselves. During user tests we compared two approaches for dealing with the conversational space: one focused on using NLP for understanding users' sentences and the other restricting what users can say through the use of *cards* and *quick replies* indicating what users can do next. Overall, users were more impressed with the NLP approach but considered the more restricted chatbot easier to use. Nevertheless, a mixed approach could yield good results, letting users say what they want but also using visual aids so they know what they can do next.

- **Demonstration of the applicability of the Semiotic Inspection Method on chatbots**. We showed that SIM can be used to inspect chatbots with no need for modifications to the method, but some issues must be taken into account regarding signs classification into metalinguistic, static, and dynamic: a semantic and contextual analysis is necessary for classifying signs as metalinguistic or static, as they share the same signification system. The open communication space can also be dealt with by having more than one inspector (even when not making the scientific application of SIM), as well as compiling a list of sentences and actions to be carried over inspections. These points were also discussed, contributing to chatbot and HCI research. Although we only analyzed text-based chatbots on Messenger, our approach could be adapted for other types of conversational interfaces as future work.

- **Demonstration of the usage of Pragmatics aspects for analyzing chatbots**. We analyzed sentences of three chatbots in the light of Speech Acts, Cooperative Principle, and Politeness Principle. With few considerations, we have shown that these Pragmatics aspects can also be used for generating a better account of the strategies, sign classes, and communications breakdown found in chatbots. During our user tests, some of the points users talked about were related to the maxims, e.g.: confusing sentences sent by the chatbot (Cooperative Principle), or how the chatbot politely took the blame for errors (Politeness Principle). This approach also can be used by chatbot designers for improving their conversation design, as well as analyzing possible breakdowns. Further investigation is needed on cases of messages that make sense in one context but not in others. Chatbots from other cultures can be analyzed to check if they treat some aspects differently, especially regarding the Politeness Principle. Another possible future work is to better instrumentalize these principles (i.e. generating a set of recommendations or aspects to be considered based on these principles) so they can be used by designers when developing or analyzing their chatbots.

- **Implementation of a chatbot using the found strategies**. We created a chatbot called *Cinemito* which reimplements functions from an existing chatbot using a different approach to its design and including as many of the strategies for conveying features as possible. Both chatbots were instrumental in the user-tests we then performed.

- **User-tests comparing two approaches for chatbots**. We performed a user study comparing two prototype chatbots that used two different approaches for performing the same tasks. Although it was a small study, we still were able to reach interesting results that complement and help consolidate the strategies and sign classes, as well as the communicative elements of the chatbots. Interesting ways to complement these results include performing tests using chatbots with better conversational capacity (as the ones we used were prototypes with very narrow focus), tests with more participants, especially from different backgrounds, instead of just tech specialists.

This work included a series of chatbots in its analysis, nevertheless, new studies considering different chatbot platforms, as well as interaction paradigms (more focused on natural language, on guiding the user, or a combination of both), and other domains (we focused mainly on news) are an interesting way of consolidating, or even expanding, our results.

The inspections in this work focused on strategies for conveying features to users. There may be other strategies focusing on other aspects, such as dealing with communication breakdowns, user onboarding, or convincing users to sign up for services. Inspections of other chatbots can also shed light on strategies for these ends. When sufficient inspections are made and the sign classes and strategies (for various ends) are mature enough, guidelines and interaction patterns can be derived from them. This work is a step in that direction.

The prototype chatbot we made for the user tests was very simple. It would be interesting to develop a more complete chatbot based on the results of this work (sign classes, strategies, and Pragmatics' principles) and evaluate both the resulting chatbot as well as the developing process itself (that is, discovering what is more or less difficult to follow during development, among others). Maybe by developing multiple chatbots, each applying different aspects of this work could lead to a more complete comparative study.

Gathering perspectives from other stakeholders (such as other researchers interested in chatbots) on the use of the sign classes and communicative strategies, as well as the considerations of Pragmatics aspects, during the design of chatbots (in the academic field or not) could lead to several improvements on the findings of this work. Besides that, it also would be interesting to have designers using the results of this work to guide them when developing chatbots, so we can investigate their use out of the academy and better support designers in real-life situations.

Finally, another way to further complement this work is by analyzing the interaction recordings during the user tests through the light of other Semiotic Engineering methods, such as the Communicability Evaluation Method.

Bibliography

Austin, J. L. (1962). *How to Do Things with Words*. Oxford University Press, Oxford.

Bento, L. F. H., Prates, R. O., and Chaimowicz, L. (2009). Using Semiotic Inspection Method to Evaluate a Human-Robot Interface. In *2009 Latin American Web Congress*, pages 77–84, New York, NY, USA. IEEE.

Bobrow, D. G. (1964). Natural Language Input for a Computer Problem Solving System. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA.

Brown, P. and Levinson, S. C. (1978). Universals in language usage: Politeness phenomena. In Goody, E. N., editor, *Questions and politeness: strategies in social interaction*, pages 56–311. Cambridge University Press.

Brown, P. and Levinson, S. C. (1987). *Politeness: Some universals in language usage*. Cambridge University Press.

Candello, H. and Pinhanez, C. (2016). Designing Conversational Interfaces. In *Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais - IHC 2016*, volume C - Livro dos Tutoriais. Sociedade Brasileira de Computação - SBC, Porto Alegre, RS, Brazil, 15th edition.

Candello, H., Pinhanez, C., and Figueiredo, F. (2017). Typefaces and the Perception of Humanness in Natural Language Chatbots. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3476–3487, New York, NY, USA. ACM.

Chaves, A. P. and Gerosa, M. A. (2019). How should my chatbot interact? A survey on human-chatbot interaction design. *CoRR*, abs/1904.02743.

Clark, L., Pantidi, N., Cooney, O., Doyle, P., Garaialde, D., Edwards, J., Spillane, B., Gilmartin, E., Murad, C., Munteanu, C., Wade, V., and Cowan, B. R. (2019). What Makes a Good Conversation?: Challenges in Designing Truly Conversational Agents. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 475:1–475:12, New York, NY, USA. ACM. event-place: Glasgow, Scotland Uk.

Corrêa, L. P. D., Coutinho, F. R. S., Prates, R. O., and Chaimowicz, L. (2012). Uso Do MIS Para Avaliar Signos Sonoros: Quando Um Problema De Comunicabilidade Se

Torna Um Problema De Acessibilidade. In *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems*, IHC '12, pages 47–56, Porto Alegre, RS, Brazil. Brazilian Computer Society.

de A. Barbosa, C. M., Prates, R. O., and de Souza, C. S. (2007). Identifying potential social impact of collaborative systems at design time. In Baranauskas, C., Palanque, P., Abascal, J., and Barbosa, S. D. J., editors, *Human-Computer Interaction – INTERACT 2007*, pages 31–44, Berlin, Heidelberg. Springer Berlin Heidelberg.

de S. Reis, S. and Prates, R. O. (2011). Applicability of the Semiotic Inspection Method: A Systematic Literature Review. In *Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction*, IHC+CLIHC '11, pages 177–186, Porto Alegre, RS, Brazil. Brazilian Computer Society.

de Souza, C. S. (2005). *The Semiotic Engineering of Human-computer Interaction*. MIT Press.

de Souza, C. S., Leitão, C. F., Prates, R. O., and da Silva, E. J. (2006). The semiotic inspection method. In *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems*, IHC '06, pages 148–157, New York, NY, USA. ACM.

de Souza, C. S. and Leitão, C. F. (2009). *Semiotic Engineering Methods for Scientific Research in HCI*. Synthesis lectures on human-centered informatics. Morgan & Claypool, San Rafael, CA, USA.

de Souza, C. S., Leitão, C. F., Prates, R. O., Bim, S. A., and da Silva, E. J. (2010). Can inspection methods generate valid new knowledge in HCI? The case of semiotic inspection. *International Journal of Human-Computer Studies*, 68(1–2):22–40.

D'Ávila, T. C. (2018). *Kino: an approach for rule-based chatbot development, monitoring and evaluation*. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil.

Fischer, J. E., Reeves, S., Porcheron, M., and Sikveland, R. O. (2019). Progressivity for voice interface design. In *Proceedings of the 1st International Conference on Conversational User Interfaces*, CUI '19, New York, NY, USA. Association for Computing Machinery.

Følstad, A., Nordheim, C. B., and Bjørkli, C. A. (2018). What Makes Users Trust a Chatbot for Customer Service? An Exploratory Interview Study. In Bodrunova, S. S., editor, *Internet Science*, Lecture Notes in Computer Science, pages 194–208, Cham. Springer International Publishing.

Galvão, V. F., Maciel, C., and Garcia, A. C. B. (2019). Creating chatbots to talk with humans: Hci evaluations and perspectives. In *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems*, IHC '19, New York, NY, USA. Association for Computing Machinery.

Gartner (2018). Top Strategic Predictions for 2018 and Beyond: Pace Yourself, for Sanity's Sake. `https://www.gartner.com/en/doc/3803530-top-strategic-predictions-for-2018-and-beyond-pace-yourself-for-sanitys-sake`. [Online, Accessed June, 2019].

Grand View Research (2017). Chatbot Market Size To Reach $1.25 Billion By 2025 | CAGR: 24.3%. `https://www.grandviewresearch.com/press-release/global-chatbot-market`. [Online, Accessed June, 2019].

Grice, H. P. (1975). Logic and conversation. In Cole, P. and Morgan, J., editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pages 41–58. Academic Press, New York.

Jain, M., Kumar, P., Kota, R., and Patel, S. N. (2018). Evaluating and Informing the Design of Chatbots. In *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, pages 895–906, New York, NY, USA. ACM. event-place: Hong Kong, China.

Jiang, J., Awadallah, A. H., Jones, R., Ozertem, U., Zitouni, I., Kulkarni, R. G., and Khan, O. Z. (2015). Automatic Online Evaluation of Intelligent Assistants. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 506–516, New York, NY, USA. ACM.

Kiseleva, J., Williams, K., Awadallah, A. H., Crook, A. C., Zitouni, I., and Anastasakos, T. (2016). Predicting User Satisfaction with Intelligent Assistants. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 45–54, New York, NY, USA. ACM.

Leahu, L., Cohn, M., and March, W. (2013). How Categories Come to Matter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3331–3334, New York, NY, USA. ACM.

Leech, G. N. (1983). *Principles of pragmatics*. Longman linguistics library. Longman, London.

Leech, G. N. (2014). *The pragmatics of politeness*. Oxford studies in sociolinguistics. Oxford University Press, Oxford.

Lewis, C. (1982). *Using the" thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.

Liao, Q. V., Davis, M., Geyer, W., Muller, M., and Shami, N. S. (2016). What Can You Do?: Studying Social-Agent Orientation and Agent Proactive Interactions with an Agent for Employees. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, pages 264–275, New York, NY, USA. ACM.

Lopes, A., Conte, T., and de Souza, C. S. (2019). Reducing the risks of communication failures through software models. In *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems*, IHC '19, New York, NY, USA. Association for Computing Machinery.

Luger, E. and Sellen, A. (2016). "Like Having a Really Bad PA": The Gulf Between User Expectation and Experience of Conversational Agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5286–5297, New York, NY, USA. ACM.

McTear, M., Callejas, Z., and Griol, D. (2016a). *The Conversational Interface.* Springer.

McTear, M., Callejas, Z., and Griol, D. (2016b). Evaluating the Conversational Interface. In *The Conversational Interface*, pages 379–402. Springer International Publishing.

McTear, M. F. (2017). The Rise of the Conversational Interface: A New Kid on the Block? In Quesada, J. F., Martín Mateos, F.-J., and López Soto, T., editors, *Future and Emerging Trends in Language Technology. Machine Learning and Big Data*, Lecture Notes in Computer Science, pages 38–49. Springer International Publishing.

Peirce, C. S. (1992). *The essential Peirce: selected philosophical writings.* Indiana University Press, Bloomington.

Piccolo, L. S. G., Mensio, M., and Alani, H. (2019). Chasing the chatbots. In Bodrunova, S. S., Koltsova, O., Følstad, A., Halpin, H., Kolozaridi, P., Yuldashev, L., Smoliarova, A., and Niedermayer, H., editors, *Internet Science*, pages 157–169, Cham. Springer International Publishing.

Portela, M. and Granell-Canut, C. (2017). A New Friend in Our Smartphone?: Observing Interactions with Chatbots in the Search of Emotional Engagement. In *Proceedings of the XVIII International Conference on Human Computer Interaction*, Interacción '17, pages 48:1–48:7, New York, NY, USA. ACM.

Purewal, S. J. (2016). The difference between Google Now and Google Assistant. `https://www.cnet.com/how-to/the-difference-between-google-now-and-google-assistant/`. [Online, Accessed June, 2019].

Radziwill, N. M. and Benton, M. C. (2017). Evaluating Quality of Chatbots and Intelligent Conversational Agents. *arXiv:1704.04579 [cs].*

Raymond, E. S. and Landley, R. W. (2004). The Art of Unix Usability. [Online, Accessed June, 2019].

Research and Markets (2019). Chatbot market to 2027 - global analysis and forecasts by component, deployment, and usage end-user. https://www.researchandmarkets.com/reports/4769883/chatbot-market-to-2027-global-analysis-and. [Online, Accessed June, 2019].

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*. Univ. Press, Cambridge.

Searle, J. R. (1975). A Taxonomy of Illocutionary Acts. *Language, mind, and knowledge*, 7:344–369.

Simonite, T. and Griffith, E. (2018). Facebook's Virtual Assistant M Is Dead. So Are Chatbots. https://www.wired.com/story/facebooks-virtual-assistant-m-is-dead-so-are-chatbots/. [Online, Accessed June, 2019].

Smestad, T. L. and Volden, F. (2019). Chatbot Personalities Matters. In Bodrunova, S. S., Koltsova, O., Følstad, A., Halpin, H., Kolozaridi, P., Yuldashev, L., Smoliarova, A., and Niedermayer, H., editors, *Internet Science*, Lecture Notes in Computer Science, pages 170–181, Cham. Springer International Publishing.

Thompson, C. (2002). Approximating Life. *The New York Times*. https://www.nytimes.com/2002/07/07/magazine/approximating-life.html [Online, Accessed June, 2019].

Tieryas, P. (2017). My Strange Conversations With Seaman. https://kotaku.com/my-strange-conversations-with-seaman-1820531394. [Online, Accessed June, 2019].

Valério, F. A. M., Guimarães, T. G., and Prates, R. O. (2016). Strategies for Encouraging Sharing in Social Networks for Professionals. In *Proceedings of the 15th Brazilian Symposium on Human Factors in Computer Systems*, IHC '16, pages 54:1–54:4, New York, NY, USA. ACM.

Valério, F. A. M., Guimarães, T. G., Prates, R. O., and Candello, H. (2017). Here's What I Can Do: Chatbots' Strategies to Convey Their Features to Users. In *Proceedings of the 16th Brazilian Symposium on Human Factors in Computer Systems*, IHC '17, New York, NY, USA. ACM.

Valério, F. A. M., Guimarães, T. G., Prates, R. O., and Candello, H. (2018). Chatbots Explain Themselves: Designers' Strategies for Conveying Chatbot Features to Users. *SBC Journal on Interactive Systems*, 9(3):61–79.

VB Profiles (2018). Bots Landscape Market Summary, Employees and Funding Information. https://www.vbprofiles.com/markets/bots-landscape-575eac092413c09f88000f12. [Online, Accessed June, 2019].

Walker, M. A., Litman, D. J., Kamm, C. A., and Abella, A. (1997). PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, EACL '97, pages 271–280, Stroudsburg, PA, USA. Association for Computational Linguistics.

Weizenbaum, J. (1966). ELIZA — A Computer Program for the Study of Natural Language Communication Between Man and Machine. *Commun. ACM*, 9(1):36–45.

Winograd, T. (1971). Procedures as a Representation for Data in a Computer Program for Understanding Natural Language.

Appendix A

Interview Consent Form

A.1 English

Consent Form

You are invited to participate on the research project "Chatbots and interaction" as a volunteer. This research is the responsibility of the following researchers: Francisco Albernaz Machado Valério, phone number [redacted] and Raquel Oliveira Prates, phone number [redacted].

If this Consent Form contains information that you cannot understand, any doubts can be answered by the researchers or team and, after all clarifications are given, in case you agree to participate in the study, we ask you to initial the pages and sign at the end of this document, which is in two copies: one will be given to you, and the other to the responsible researcher.

In case you disagree to participate, there will be no penalty to you, and it will also be possible to withdraw your consent at any time, also with no penalty.

This research aims to evaluate distinct styles of chatbot design and their impacts on interaction. At this stage, we will conduct interviews. In this activity, you are invited to interact with chatbots and answer questions related to you interaction with them. The activity will have the approximate duration of 20 minutes, will be recorded, and will be conducted at UFMG, at a time defined according to your preference and availability.

The risks involved in this research are considered low, but not nonexistent. There is the risk of embarrassment, discomfort, fatigue while answering questions, and breach of anonymity. To minimize these risks, it is informed what is expected from your participation, that your participation is voluntary, and that you can quit at any time. Your anonymization is guaranteed, and this information is reinforced. All assistance will be provided to you in any accident or discomfort.

There are also several benefits associated with this research. The results obtained through this research are important to advancing exploratory research about different interaction methods in conversational interfaces

The information of this research will be confidential, and will be disclosed only in scientific events and publications, without volunteer identification except for the person contacting you - i.e. other researchers will not have access to your identification, ensuring confidentiality of the volunteer participation. Data collected in this research (audio files) will be stored in computers of the Computer Science Department (DCC - UFMG) for the minimum period of five years.

You will not pay anything, nor will receive payment to participate in this research, since it must be volunteer, but it is also guaranteed the indemnity in case of damage proven to be resulting of your participation in the research, as from judicial or extra-judicial decision. If needed, the expenses for your participation will be payed by the researchers (transportation reimbursement).

I, ____, CPF ________________, signed below, agree in participating of the study "Chatbots and interaction", as a volunteer. I have been properly informed by the researcher about this research, the procedures involved, as well as the possible risks and benefits resulting from my participation in it. It was guaranteed to me that I can withdraw my consent at any moment, without any penalty to me.

Belo Horizonte, ________ of ________________ of ________.

______________________________ ______________________________

Volunteer Signature Researcher Signature

A.2 Portuguese

Termo de Consentimento Livre e Esclarecido

Você está convidado(a) para participar, como voluntário(a), da pesquisa "Chatbots e interação". Esta pesquisa é da responsabilidade dos pesquisadores: Francisco Albernaz Machado Valério, telefone para contato [censurado] e Raquel Oliveira Prates, telefone para contato [censurado].

Caso este Termo de Consentimento contenha informações que não lhe sejam compreensíveis, as dúvidas podem ser tiradas com os pesquisadores ou equipe e, apenas ao final, quando todos os esclarecimentos forem dados, caso concorde em fazer parte do estudo pedimos que rubrique as folhas e assine ao final deste documento, que está em duas vias, uma via lhe será entregue e a outra ficará com o pesquisador responsável.

Caso não concorde, não haverá penalização para você bem como será possível retirar o consentimento a qualquer momento, também sem nenhuma penalidade.

Esta pesquisa tem o objetivo de avaliar diferentes estilos e design de chatbots e seus impactos na interação. Nesta etapa realizaremos testes com usuários. Nesta atividade, você é convidado a usar chatbots e responder perguntas relacionadas à interação com os mesmos. A Atividade terá duração aproximada de 20 minutos, será gravada, e será realizada na UFMG, sendo definido um horário de acordo com a sua preferência e disponibilidade.

Os riscos envolvidos nesta pesquisa são considerados baixos, porém não são inexistentes. Existe o risco de constrangimento com alguma pergunta, desconforto, cansaço e quebra de anonimato. Para minimizar estes riscos, é esclarecido o que é esperado da sua participação, de que a sua participação é voluntária e que poderá desistir a qualquer momento sem consequências. A anonimização é garantida, e essa informação é reforçada. Toda assistência será prestada você caso haja algum acidente ou desconforto.

Existem também benefícios associados a esta pesquisa. Os resultados obtidos através da pesquisa são importantes para o avanço da pesquisa exploratória sobre diferentes métodos de interação em interfaces conversacionais.

As informações desta pesquisa serão confidenciais e serão divulgadas apenas em eventos ou publicações científicas, não havendo identificação dos voluntários, a não ser a pessoa pela entrevistadora, i.e. os demais responsáveis não terão acesso à sua identificação, sendo assegurado o sigilo sobre a participação do(a) voluntário(a). Os dados coletados nesta pesquisa (arquivos de áudio e vídeo), ficarão armazenados em computadores do Departamento de Ciência da Computação (DCC-UFMG), pelo período mínimo de 5 anos.

O(a) senhor(a) não pagará nada e nem receberá nenhum pagamento para participar desta pesquisa, pois deve ser de forma voluntária, mas fica também garantida a indenização em casos de danos, comprovadamente decorrentes da sua participação na pesquisa, conforme decisão judicial ou extrajudicial. Se houver necessidade, as despesas para a participação serão assumidas pelos pesquisadores (ressarcimento com transporte).

Eu, ____,
CPF ________________, abaixo assinado, concordo em participar no estudo "Chatbots e interação", como voluntário(a). Fui devidamente informado(a) e esclarecido (a) pelo(a) pesquisador(a) sobre a pesquisa, os procedimentos nela envolvidos, assim como os possíveis riscos e benefícios decorrentes da participação dele(a). Foi-me garantido que posso retirar o meu consentimento a qualquer momento, sem que isto leve a qualquer penalidade (ou interrupção de seu acompanhamento/ assistência/tratamento) para mim.

Belo Horizonte, ________ de ________________ de ________.

Assinatura do(a) voluntário(a)	Assinatura do(a) pesquisador(a)