Chatbots Explain Themselves: Designers' Strategies for Conveying Chatbot Features to Users

Francisco A. M. Valério*, Tatiane G. Guimarães[†], Raquel O. Prates[‡] and Heloisa Candello[§]
*[†][‡]DCC, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil
[§]IBM Research, São Paulo, SP, Brazil
{*falbernaz, [†]tatiguimaraes, [‡]rprates}@dcc.ufmg.br, [§]hcandello@br.ibm.com

***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 dealing with all the necessary Natural Language Processing. But, as chatbots use natural language as interface, their users may struggle to discover which sentences the chatbots will understand and what they can do. Because of that it is important to support their 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 as a result we were able to identify a series of strategies used by the analyzed chatbots for conveying their features to users. We then consolidate these findings by analyzing other chatbots. Finally, we discuss the use of these strategies, as well as challenges for designing such interfaces and limitations of using SIM on them.**

***Index Terms*—Chatbots, Conversational Interfaces, Evaluation, Semiotic Engineering, Semiotic Inspection Method.**

I. Introduction

SINCE the sixties we have been able to talk to computers through intelligent software such as STUDENT [1] and ELIZA [2]. That kind of software allows for interaction using text-based natural language as input and it produces natural language output [3]. Since then, chatbot technology has greatly advanced. Chatbot APIs for Online Social Networks and messenger applications (e.g. Facebook Messenger and Telegram), and Software-As-Service platforms (for creating chatbots with little to no programming required, such as IBM's Watson Conversation Service and Wit.ai) helped pushing chatbots popularity in recent years.

Chatbots are different from traditional graphical user interfaces. They use text-based natural language [3] instead of other graphical elements such as buttons and menu bars other systems use. That way, they 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.

Little is known about the best strategies to use when conveying chatbots' features to users. Most evaluation methods used to assess text-based chatbots focus on specific components [4], such as benchmarks for Natural Language Processing and annotated results for queries made to the chatbot [5]. There are only a few qualitative evaluation methods for text-based chatbots evaluation [4].

The Semiotic Inspection Method (SIM) [6] is a qualitative evaluation method based on Semiotic Engineering [7] that assesses systems' communicability. Communicability is the system's property related to its ability to communicate efficiently (in an organized and resourceful manner) and effectively (that achieves the desired result) the designer's intentions and underlying principles to users [7], [8]. In a traditional system, SIM results would allow specialists to identify communicative strategies used by the system designer. Nevertheless, there is no record of this method being used in the context of chatbots.

In this paper, our goal is to analyze communicative strategies used by popular chatbots to convey their features to users. Our study is divided in two rounds of inspections. In the first round, we applied SIM in a bottom-up approach and, through the analysis of three chatbots, we identified the communicative strategies used by designers to convey the chatbots' features to users as well as classes of visual cues used on the chatbots [9]. 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 (not using SIM). This second round comprehended a larger set of ten chatbots, enabling us to have a broader view and to consolidate our previous findings.

As result, we were able to consolidate the list of sign classes and strategies that emerged from the first round of inspections. Furthermore, one more strategy was identified in the second round and added to the set of strategies. These results can support chatbot designers (as well as chatbot platform's designers) when deciding how and when to communicate chatbots' features to users. We discuss two main challenges for designers: the openness of the communication space and the hidden structure that are part of the chatbots' essence. We also discuss some limitations and challenges regarding the use of SIM on conversational interfaces.

This paper is organized as follows. First we present related work on chatbots and SIM. Next, we explain the methodology used in this work, including the premises we assumed when using SIM on chatbots, and the first and second rounds of inspections. We then show the results of the first round, namely the sign classes and strategies for conveying features identified. Then, we show the results of the consolidation round, discussing both the presence of the sign classes and strategies on other chatbots, as well as designers' choices taking them into account. Later, we discuss design considerations, as well as perceived challenges and limitations of SIM in this context. Finally, we present our conclusions and future works.

II. Related Works

This section presents relevant work on text-based chatbot evaluation. It then briefly introduces Semiotic Engineering theory, which is the basis for the Semiotic Inspection Method, and then it presents the method itself.

A. Chatbots

Chatbots have been around for a long time and had a recent boost in popularity. However, there is little research supporting their design. Recent works in this direction explored the influence of typefaces in users' perception of humanity in chatbots [10]; and how different chatbot's behaviors (e.g. using social cues, humor, and timing) may affect user engagement [11].

In traditional graphical user interfaces, designers use signs[1] 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 [3].

There are many 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 [4].

In their systematic literature review, Radziwill and Benton [13] make 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 [13] also list quality assessment approaches found in literature, such as the PARADISE framework [14], among others, and propose a new approach based on the identified quality attributes.

However, to the best of our knowledge, there is no 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 first 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. To do so, we decided to use the Semiotic Inspection Method, since it evaluates the communicability of a system and it has been shown that it can be applied to a wide range of different contexts and technologies [15].

In the next subsection, we present the Semiotic Inspection Method, and a brief introduction of its basis: the Semiotic Engineering Theory.

B. 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 designers 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 has been defined as communicability [7], [8]). Thus, in order to assess this property, communicability evaluation methods have been proposed. One of these methods is the Semiotic Inspection Method [6], [16]. 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 [6].

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 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 on this type of sign only. Metalinguistic signs "explicitly communicate to users the meanings encoded in the system and how they can be used" [8], 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" [8], 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

[1] In the present work we use the term "sign" in the Peircian sense, of anything that may mean something to someone [12].

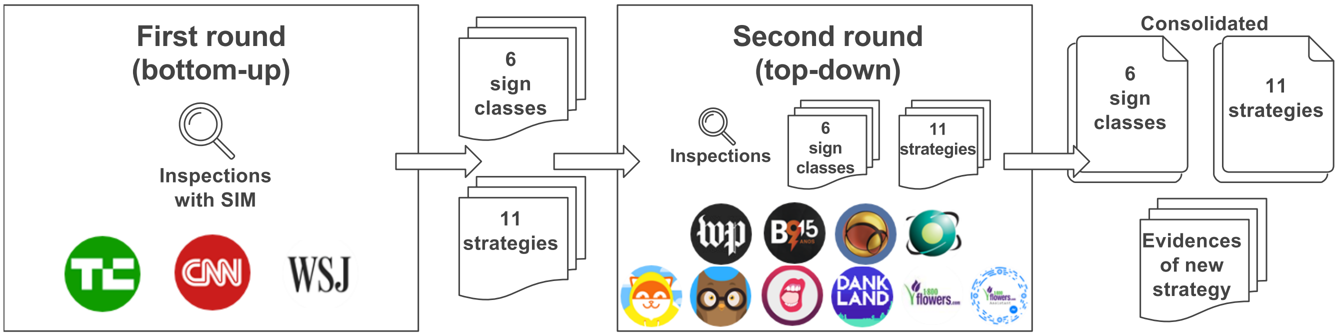


Fig. 1. Methodology overview.

bound to temporal and causal aspects of the interface, namely, to interaction itself" [8], 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 [16].

These first three steps generate a segmented analysis of the interface, whereas the following two steps integrate them. In step **(iv)**, the specialist contrasts the three metacommunication messages reconstructed during the segmented analysis and compares them looking for inconsistencies and potential problems. Finally, in step **(v)**, 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 [16]. 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.

III. Methodology

This section describes the methodology used in this work in order to identify and evaluate the strategies being used to convey chatbot features to users.

Our 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 and strategies that had not been identified yet. Fig. 1 shows an overview of the adopted methodology.

The choice of SIM to inspect the chatbots was motivated by the fact that the method allows for a systematic analysis of the designers' metacommunication [16], and thus does not depend on specific technologies or domains [15]. However, as it had not been applied before to conversational interfaces, in this section we explain the challenges faced in its 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.

A. *Premises Regarding SIM Applicability*

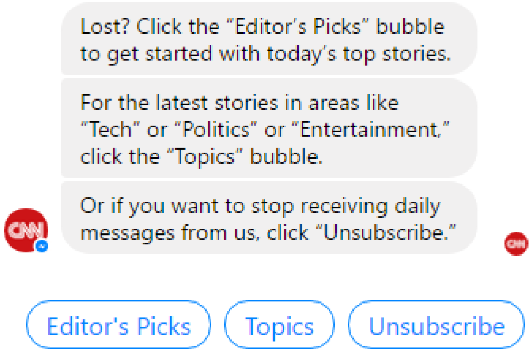
SIM focuses on communicative aspects of the intended metamessage sent from designers to users. By supporting the reconstruction of the metamessage from the system's designers, SIM allows for the identification of communicative strategies used by the designers to communicate their design intents and principles [8]. It has been shown that it can be applied to variety of contexts and technologies [15]. For instance, SIM has been used on educational software [15], human-robot interface [17], the audio aspect of video games [18], and online social networks [19]. 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 is the first time it is applied in this context. Therefore, we had to adopt some premises for the inspections.

Different from other domains and technologies SIM has been applied to, chatbot interaction is based mainly on natural language message exchange, simulating a human-to-human conversation. Therefore, our first challenge in order to apply SIM to this context was to check whether the Semiotic Engineering's definitions of what should be considered metalinguistic, static, and dynamic signs held for 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 [6], [8], [16]. 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, 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. 2 illustrates an example of a metalinguistic sign on the CNN's chatbot in which it informs the user about how to access some of its features. In addition, as in the application of SIM in other contexts, we also considered news 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.


Lost? Click the "Editor's Picks" bubble to get started with today's top stories.

For the latest stories in areas like "Tech" or "Politics" or "Entertainment," click the "Topics" bubble.

Or if you want to stop receiving daily messages from us, click "Unsubscribe."

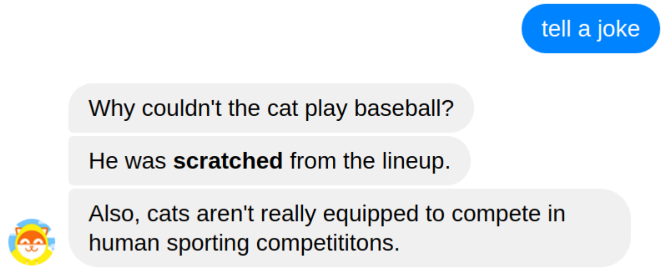
Fig. 2. Example of metalinguistic sign on the CNN's chatbot.

Utterances not considered metalinguistic signs (that is, about topics other than the chatbot itself or its features) were classified as static signs. Fig. 3 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*[2]) were also considered static signs.

Dynamic signs are defined as signs that are bound to temporal and causal relations, and represent the interaction itself [6], [8], [16]. 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 are bound to temporal or causal relations.

B. 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


tell a joke

Why couldn't the cat play baseball?

He was **scratched** from the lineup.

Also, cats aren't really equipped to compete in human sporting competititons.

Fig. 3. Example of static sign on the Poncho chatbot.

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.

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 ChatBottle Awards 2017[3]. The selected chatbots were: TechCrunch[4], CNN[5], and the Wall Street Journal[6] (WSJ) chatbots. Although all three chatbots are news-related, they focus on different contents. 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 [16]. 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 all of the chatbot, for the analyzed chatbots did not have many different functions.

The inspections were performed by the first two authors of this work, who already 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 the other authors. These inspections were conducted during June 2017.

As a result of these inspections we were able to identify six sign classes used as visual cues on the chatbots' messages and 11 strategies related to presenting the chatbots' features to users. These results are shown in the IV - "First Round:

[2] The classes of visual elements identified in our analysis will be described in detail in subsection IV-A – "Sign Classes".

[3] https://chatbottle.co/awards/1st-chatbottle-awards-2017-winners, last access on Jan/2018.

[4] https://www.messenger.com/t/techcrunch, last access on Jan/2018.

[5] https://www.messenger.com/t/cnn, last access on Jan/2018.

[6] https://www.messenger.com/t/wsj, last access on Jan/2018.

Results” section: the visual classes are shown in IV-A - “Sign Classes” subsection, while the strategies are explained in IV-C - “Strategies For Conveying Features” subsection.

C. 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 emerged. 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 (news) and the others distributed in different domains, namely weather, entertainment, sales, environment, and feminism.

In each chatbot we preformed a systematic inspection, registering which sign classes and strategies (previously identified) they used and how, and if any other sign or strategy that had not been identified in the the first stage of our research emerged. Notice 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[7]. Next we list each one of the 10 selected chatbots, presenting, their name, abbreviation, URL, their domain and quick description, and language.

- **Washington Post (WP)**[8]: a news chatbot that focuses on political stories from the USA in English;
- **Brainstorm 9 (B9)**[9]: a Brazilian news chatbot that focuses on news about communication, culture, and media in Portuguese;
- **UOL Notícias (UOL)**[10]: another Brazilian news chatbot with a wide range of news topics in Portuguese;
- **BOL (BOL)**[11]: 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)**[12]: a very irreverent chatbot that informs users about the weather and tells a lot of jokes in English;
- **Smokey (SMO)**[13]: 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)**[14]: a Brazilian chatbot that focuses in keeping users up to date on feminist matters in Brazil in Portuguese;
- **Dankland (DNK)**[15]: a chatbot that can create *memes* out of figures the user sends to it in English;
- **1-800-Flowers.com (18F)**[16]: chatbot related to the *1-800-Flowers.com* online store, focusing on customer service in English; and
- **1-800-Flowers.com Assistant (18FA)**[17]: 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 first author, who also performed the inspections during the first round. The results were then discussed with the other authors. The results of this stage of the research are presented in section V - “Second Round: Findings Consolidation”.

IV. First Round: Results

This section presents and discusses the results of the first round of analyses in which three chatbots were inspected with the goal of identifying 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 to their use, as well as the communicative strategies adopted by designers.

A. Sign Classes

As mentioned before, chatbots differ from traditional graphical user interfaces specially 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 sign 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 a white text if the message was sent by the user. Fig. 4 shows a simple message sent by the TechCrunch chatbot as response to another simple message sent by the user.

[7] The chatbots were identified in a number of 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.

[8] https://www.messenger.com/t/washingtonpost, last access on Jan/2018.

[9] https://www.messenger.com/t/brainstorm9, last access on Jan/2018.

[10] https://www.messenger.com/t/UOLNoticias, last access on Jan/2018.

[11] https://www.messenger.com/t/BOL, last access on Jan/2018.

[12] https://www.messenger.com/t/hiponcho, last access on Jan/2018.

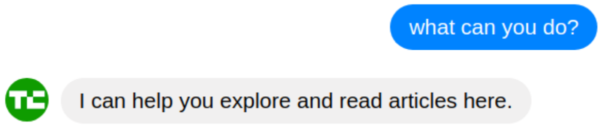
[13] https://www.messenger.com/t/smokeybot, last access on Jan/2018.

[14] https://www.messenger.com/t/beta.feminista, last access on Jan/2018.

[15] https://www.messenger.com/t/dankland, last access on Jan/2018.

[16] https://www.messenger.com/t/1800flowers, last access on Jan/2018.

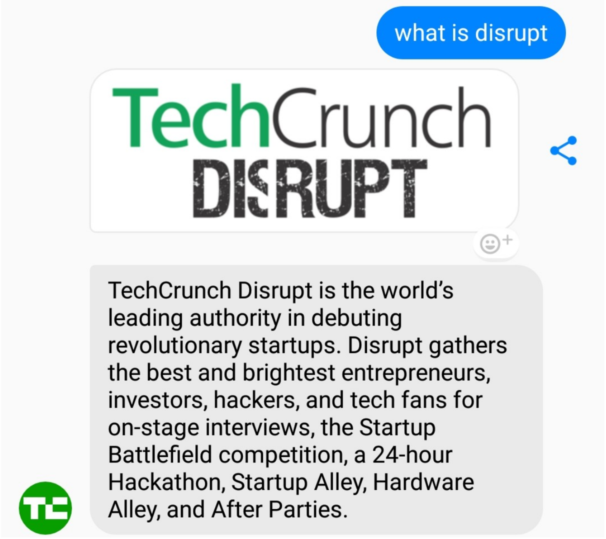
[17] https://www.messenger.com/t/1800FlowersAssistant, last access on Jan/2018.


what can you do?

I can help you explore and read articles here.

Fig. 4. TechCrunch's simple message example.

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 itself with a rounded border and a subtle grey 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 grey arrow pointing up on the web version. If the image was sent by the chatbot, there will be a small grey 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 grey emoji will not be shown. Fig. 5 shows the TechCrunch chatbot replying a simple image followed by a simple message on Messenger App.


what is disrupt

TechCrunch Disrupt is the world's leading authority in debuting revolutionary startups. Disrupt gathers the best and brightest entrepreneurs, investors, hackers, and tech fans for on-stage interviews, the Startup Battlefield competition, a 24-hour Hackathon, Startup Alley, Hardware Alley, and After Parties.

Fig. 5. TechCrunch's message containing a simple image.

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 him/herself, 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. Facebook Messenger represents suggestions with a blue outline. Fig. 6 shows suggestions of some stories the CNN chatbot can show the user: "Editor's Picks", "News", "Politics", and "Business".

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 other 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 a button with a blue text. Fig. 7 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. 8 shows an example of a card with the topic "GOOG" (a ticker symbol on the stock market) and the option "Stop Following", whereas Fig. 9 shows an image of

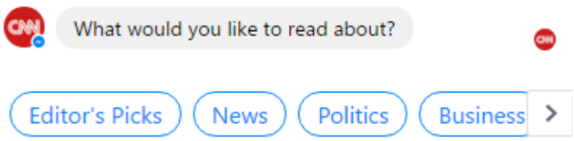
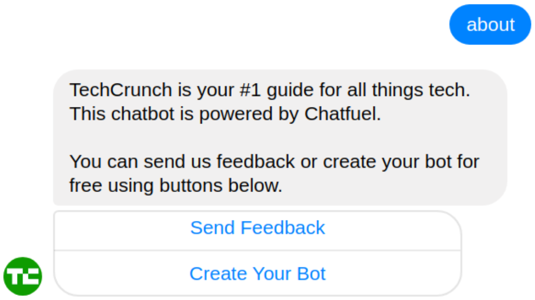

What would you like to read about?

Fig. 6. CNN's suggestions example.


about

TechCrunch is your #1 guide for all things tech. This chatbot is powered by Chatfuel.

You can send us feedback or create your bot for free using buttons below.

Fig. 7. TechCrunch's card with no associated topic nor image.

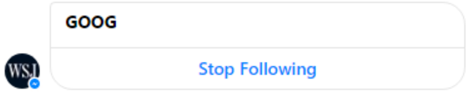
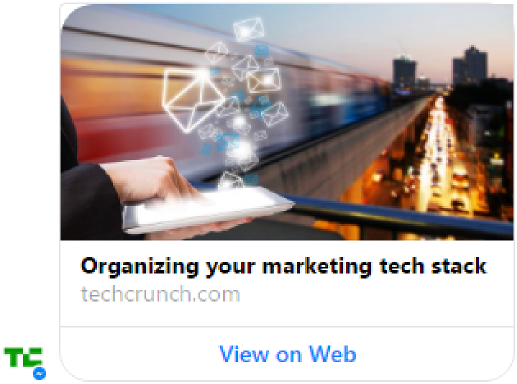


Fig. 8. WSJ's card with topic (GOOG) and no image.


Organizing your marketing tech stack

techcrunch.com

Fig. 9. TechCrunch's card with an image.

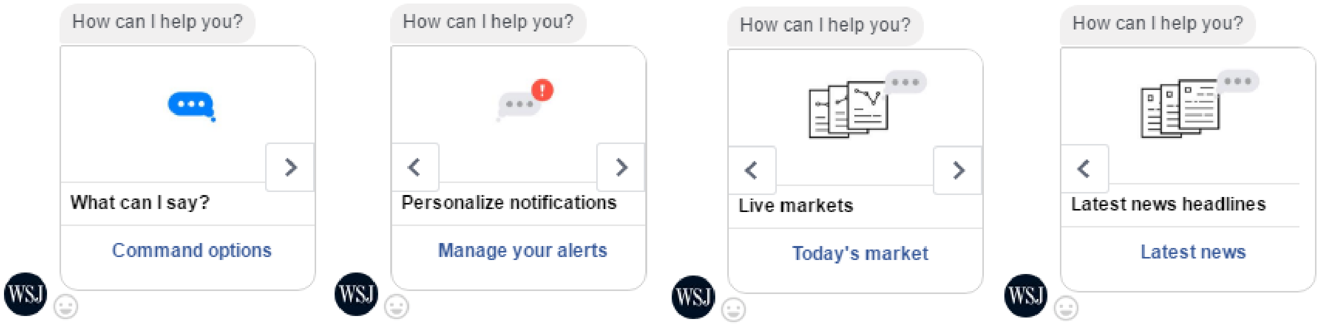


Fig. 10. WSJ's main menu carousel.

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. 10 shows the carousel WSJ uses to show its main menu: a set of cards, each with their 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 App.

C6 – Persistent Menu: a set of buttons the user can access at any time. Fig. 11 shows the persistent menu for CNN on the Messenger App interface, while Fig. 12 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.

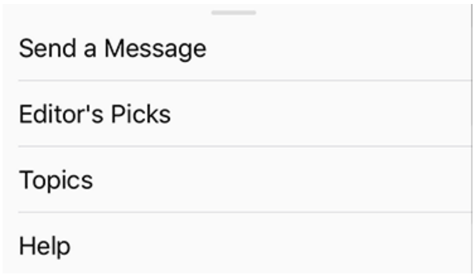


Fig. 11. CNN's persistent menu on mobile app.

B. Sign Classes Considerations

The first two sign classes presented (Simple text 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 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 usual graphical interface, and and less as natural language interaction experience.

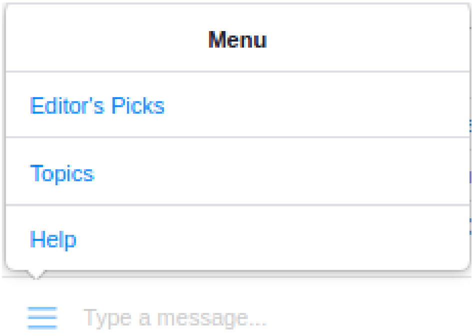
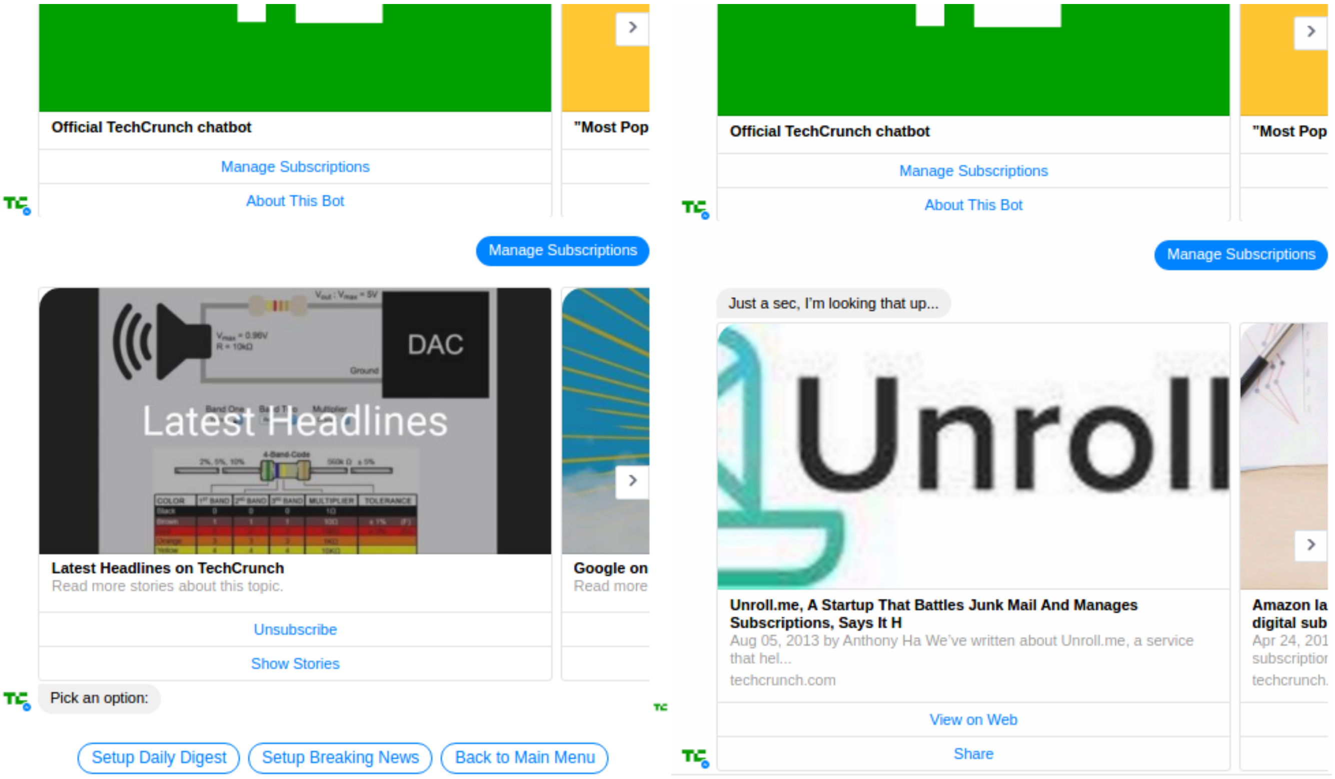


Fig. 12. CNN's persistent menu on Messenger web interface.

Once the user selects a predetermined action (a suggestion or a button in a card or in the persistent menu), Facebook Messenger represents it as if the user had typed it him/herself: 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. 13 shows an example of this: on the left side we can



Official TechCrunch chatbot

"Most Pop

Latest Headlines on TechCrunch

Read more stories about this topic.

Google on

Read more

Pick an option:

Official TechCrunch chatbot

"Most Pop

Just a sec, I'm looking that up...

Unroll.me, A Startup That Battles Junk Mail And Manages Subscriptions, Says It H

Aug 05, 2013 by Anthony Ha We've written about Unroll.me, a service that hel...

techcrunch.com

Amazon la

digital sub

Apr 24, 201

subscriptior

techcrunch.

Fig. 13. 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'.

see a dialog in which the user selects a suggestion ("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 exactly the same of 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 card after an option is selected and, thus, on 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, on their turn, would be preferable if designers want to allow users to have the possibility to explore the various options, as users could can 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 to an external link, it shows a URL in gray below the black title (as in Figures 9 and 13). These Figures also show how TechCrunch uses buttons on cards as redundancies to the link on the image. However, WSJ does not adopt this strategy. When it uses images with links on cards, it does not associate redundant buttons to them. Thus, users may not notice or understand that there is a link associated to 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. 12 shows the icon on the bottom left). On the App version, the menu can be made always accessible by default (Fig. 11), 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 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. 11), 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. 14), distinguishing it from the other options on the persistent menu.

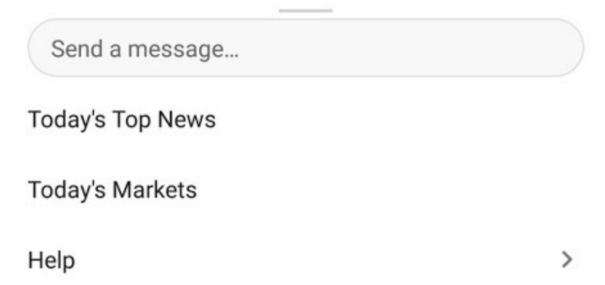


Fig. 14. WSJ's persistent menu on messenger app on January 2018

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. 14 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. 10, 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 the other chatbots adopt this strategy.

C. 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.

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 consider 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 her/him 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 include 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 "something else" button on all cards (CNN), that 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 in a different way. It also shows a "more stories" button on the last card of a story's carousel. Nevertheless, it presents *quick reply* suggestions 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 temporal constrains 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 him/her 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 on 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[18], unveiling a list of the chatbot's features that may function as shortcuts so the user does not have to type a sentence.

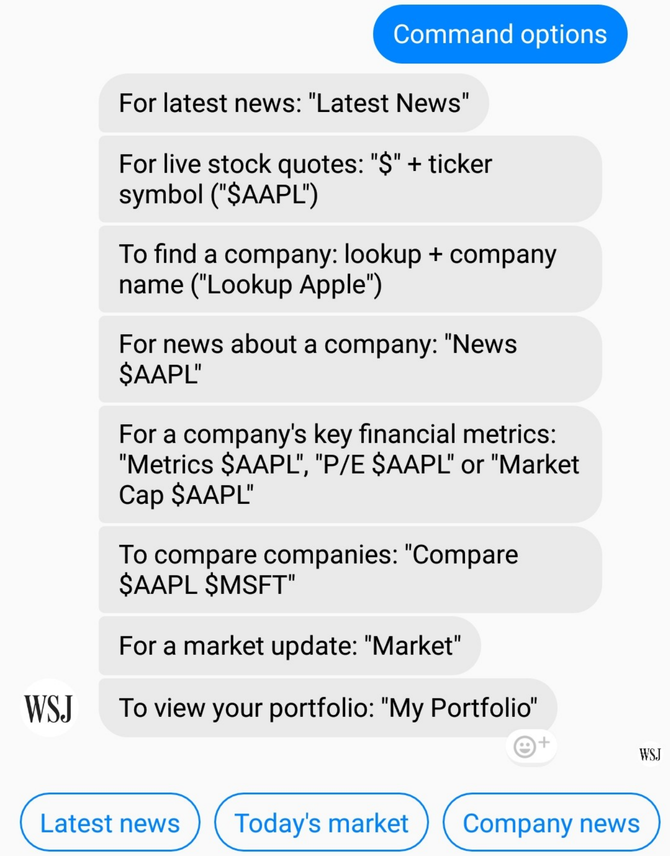
All the chatbots use this feature on Messenger App, i.e. the menu stays visible from the start, and the user has to select the option "Send a message" to be able to type a message to the chatbot.

S5 – Sending the main menu with main features as message.[19] 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 on 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. 10) 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 on Fig. 15, 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.



Command options

For latest news: "Latest News"

For live stock quotes: "$" + ticker symbol ("$AAPL")

To find a company: lookup + company name ("Lookup Apple")

For news about a company: "News $AAPL"

For a company's key financial metrics: "Metrics $AAPL", "P/E $AAPL" or "Market Cap $AAPL"

To compare companies: "Compare $AAPL $MSFT"

For a market update: "Market"

To view your portfolio: "My Portfolio"

Fig. 15. WSJ's command list on mobile app.

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. 16.

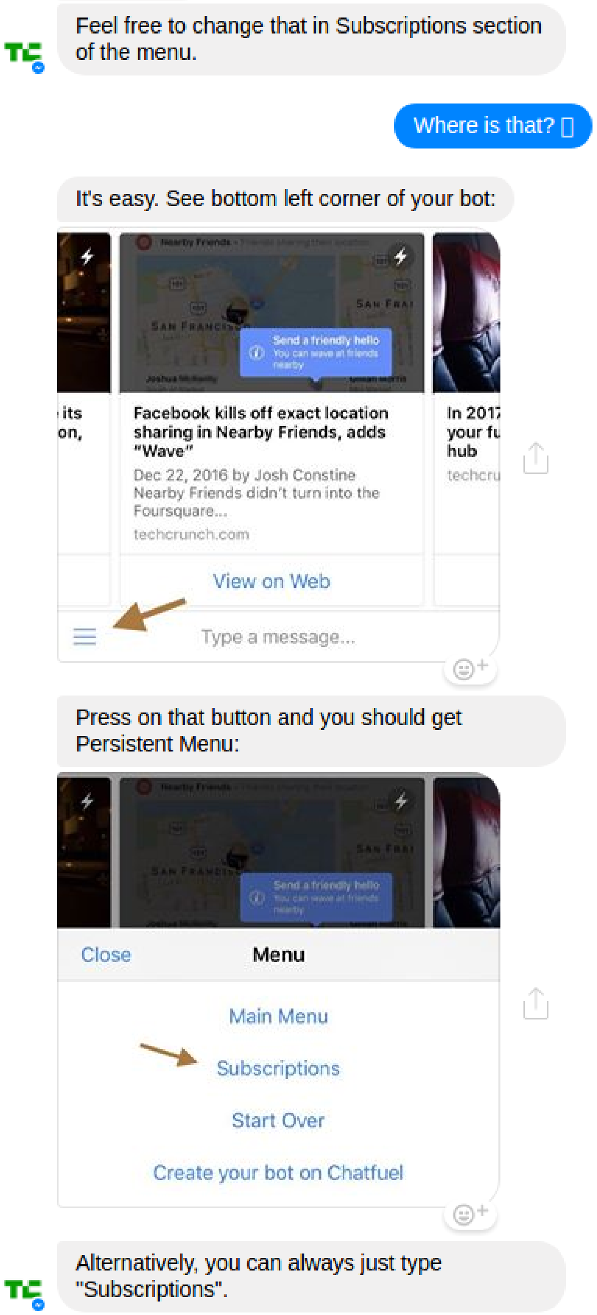
S8 – Showing the main menu or the most frequent features when the user asks for help. This strategy was identified on CNN and WSJ chatbots. It consists of showing the main menu as a response to a message from the user asking for help. Usually users ask for help when they do not know what to do next or when they do not know how to do something they want. Therefore, showing the menu 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.

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 can include: (i) the user does not know how to access some feature of the chatbot; or (ii) he/she 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

[18] 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**.

[19] The name of the S5 strategy was changed from "Having a main menu with main features" (on [19]) 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" in order to differentiate from the S4's persistent menu.



Feel free to change that in Subscriptions section of the menu.

Where is that? ▯

It's easy. See bottom left corner of your bot:

Facebook kills off exact location sharing in Nearby Friends, adds "Wave"
Dec 22, 2016 by Josh Constine
Nearby Friends didn't turn into the Foursquare...
techcrunch.com
View on Web
Type a message...

Press on that button and you should get Persistent Menu:

Close
Menu
Main Menu
Subscriptions
Start Over
Create your bot on Chatfuel

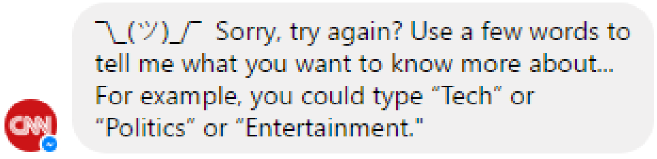
Alternatively, you can always just type "Subscriptions".

Fig. 16. TechCrunch's contextual help about the subscriptions menu.

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. 17). 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 it to the user and will show the main menu carousel (Fig. 10).



¯_(ツ)_/¯ Sorry, try again? Use a few words to tell me what you want to know more about... For example, you could type "Tech" or "Politics" or "Entertainment."

Fig. 17. CNN's example of S9.

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 recently been made possible by an update to the Facebook Messenger platform[20] and it is only available on the Messenger app, not on the web version. All of the inspected chatbots used this strategy when accessed through the smartphone app. As expected, this strategy was not available while chatting on the Messenger website.

S10 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 on 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**), on the main menu (**S5**), and on 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 on both persistent

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

and main menus (**S4** and **S5**). An example of a less advertised feature is the companies comparison, which is not present on any menu of the chatbot. The user must type “compare” followed by the desired companies ticker symbols to see a comparison chart. The command for comparing companies is only listed on the “What can I say” option in the main menu.

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. Just for an overall view of how many of the chatbots adopt each strategy:

- Five strategies (45%) are used by all three chatbots – **S3** (Suggesting next actions to the user), **S4** (Having a persistent menu with main features), **S5** (Sending the main menu with main features as message), **S10** (Showing the persistent menu instead of a text-input box), and **S11** (Highlighting the most important feature);
- 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 small 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 main features when user says something the chatbot cannot understand);
- 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%), with **S7** as the only strategy not used.

V. Second Round: 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 SIM application. It is organized in two subsections, one that describes which sign classes and strategies were found in which chatbots and our conclusions of 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.

A. Sign Classes and Strategies Presence in Other Chatbots

The results of our analysis are compiled on Tables I and II, 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 I, II, and III are the following: TC (TechCrunch), CNN, WSJ (Wall Street Journal), WP (Washington Post), B9 (Brainstorm 9), UOL (UOL Notícias), BOL, PNC (Poncho), SMO (Smokey), BET (Beta), DNK (Dankland), 18F (1-800-Flowers.com), and 18FA (1-800-Flowers.com Assistant).

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[21]. 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.

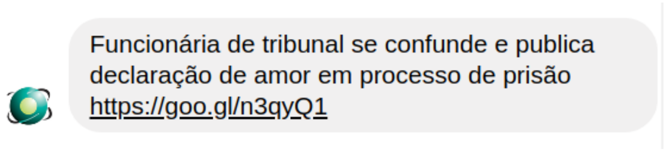
Table I shows the results of the analyses of the six sign classes over the 13 examined chatbots (three from the initial SIM inspections plus ten from the consolidation). In the Table, an “O” indicates that the sign class (rows) was present in the chatbot (columns), while a “.” means otherwise. The columns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

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, that 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 Fig. 18), 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. 9 shows an example a card presenting a piece of news on TechCrunch).

Also, Table I shows that 4 out of the 10 chatbots inspected in the second round use all of the sign classes, as does TechCrunch.

Finally, during the second round of inspections, we did not identify any signs used by the chatbots that would represent a new sign class.



Funcionária de tribunal se confunde e publica declaração de amor em processo de prisão https://goo.gl/n3qyQ1

Fig. 18. BOL chatbot’s simple message with a news story.
Translation: *Court employee publishes love declaration in sentence of arrest by mistake https://goo.gl/n3qyQ1.*

Regarding the strategies for conveying features, Table II compiles the results of the inspections looking for evidences of the strategies on the chatbots. As in Table I, the rows represent the strategies (**S1** through **S11**), the columns show the chatbots in the order the inspections took place, the last column shows

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

TABLE I
SIGN CLASSES CONSOLIDATION

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

TABLE II
STRATEGIES CONSOLIDATION

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

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 small tutorial during first messages), which was used by three chatbots (TechCrunch, WSJ and B9).

No strategy were used by all the chatbots. Nonetheless, some were much more popular than others. **S3** (Suggesting next 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** (Sending the main menu with main features as message), 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 [9] 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 a number of 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 an evidence of their adoption of **S9** (Showing the main menu or main features when user says something the chatbot cannot understand). 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 III 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 that we considered users might utter when they want to ask the chatbot for help. In each cell, an "O" 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 an "O*" and are evidences of the adoption of **S9** (Showing the main menu or main features when user says something the chatbot cannot understand) by the chatbots.

Notice that the first five rows represent sentences or words that at least one chatbot understands as being a call for help. It is interesting that 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[22].

It is also worth noting that the 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. 10), 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

[22] 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).

TABLE III
FEATURES ON SENTENCES

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

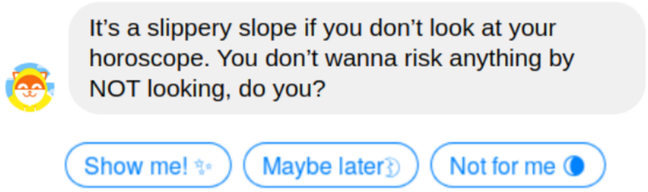

It's a slippery slope if you don't look at your horoscope. You don't wanna risk anything by NOT looking, do you?

Fig. 19. Poncho chatbot actively offering a feature.

after the inspection, Poncho sent us a message offering us to take a look at our horoscope. Prior to that, we did not subscribe for any kind of horoscope service in Poncho, so we concluded that Poncho was actively taking the initiative to offer one of its features. Fig. 19 shows the message Poncho sent us to offer the horoscope. Thus, this could be an 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.

B. 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 can result in potential communication breakdowns. Fig. 20 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 he/she wants to receive the news digest. Only after that, does the chatbot inform the user that he/she can also choose to subscribe to the other option (in this case, "Mais lidas" – "Most read"), as shown on Fig 21. When presented with the *quick replies* as in Fig. 20, one could reasonably think that it is only possible to select one of the two options for digest: 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.

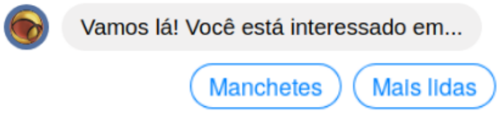
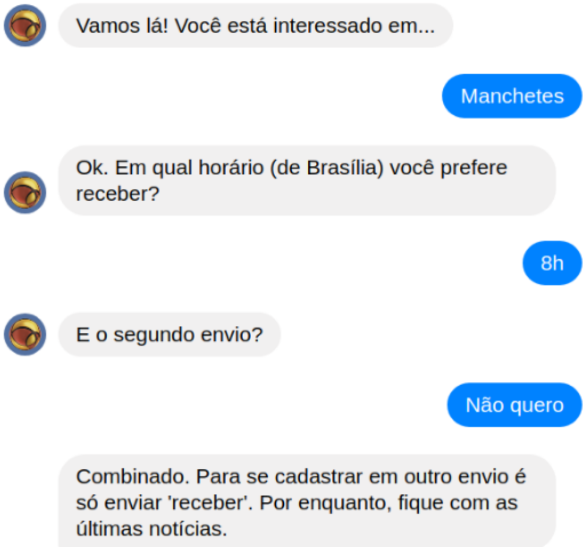

Vamos lá! Você está interessado em...

Fig. 20. Suggestions on the *UOL Notícias* chatbot.
Translation: –*Let's go! You are interested in...*
-Headlines -Most read


Vamos lá! Você está interessado em...

Manchetes

Ok. Em qual horário (de Brasília) você prefere receber?

8h

E o segundo envio?

Não quero

Combinado. Para se cadastrar em outro envio é só enviar 'receber'. Por enquanto, fique com as últimas notícias.

Fig. 21. Follow-up on the suggestions on the *UOL Notícias* chatbot.
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.*

As discussed in section IV-C - "Strategies For Conveying Features", while explaining **S1**, subscriptions to daily messages can help creating 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, it 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 Washington Post chatbot. WP does not offer any kind of subscription feature, instead relies on the user to ask it for the “Top stories”.

Another interesting fact about the Washington Post chatbot is that it only understands three commands: “Top stories”, “Contact”, and “Help”. If the user sends any message different from those three, the chatbot replies with a message stating it can only respond to those three commands (Fig. 22 shows that message). While that is a clear example of **S9** (Showing the main menu or main features when user says something the chatbot cannot understand), 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 (Fig. 22), 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.

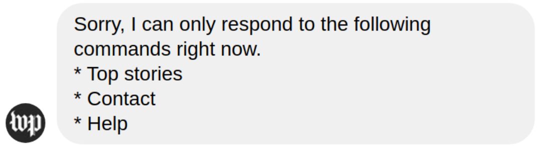
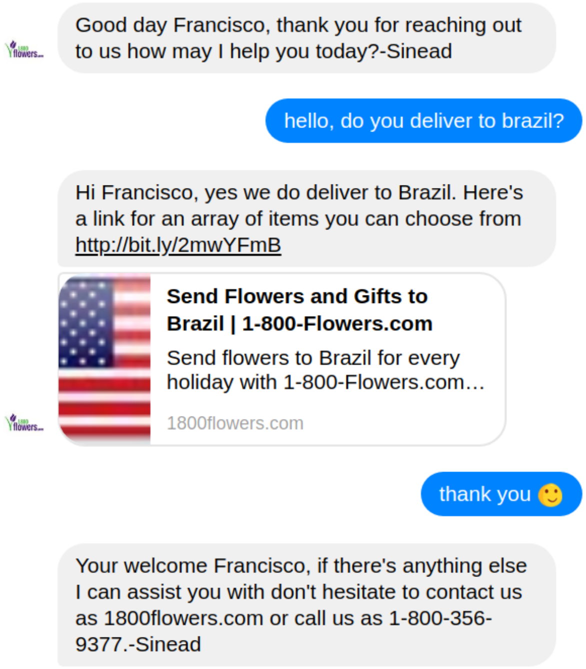

Sorry, I can only respond to the following commands right now.
* Top stories
* Contact
* Help

Fig. 22. Washington Post chatbot’s reply when it cannot understand the user’s utterance.

Beta chatbot was found to use only one strategy: **S3** (Suggesting next 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. However, if the user tries to type his/her 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 menu 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 to 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 sends a message to 18F, after a while, a real person will answer the user in name of the chatbot. While it is a great way of ensuring that the user will not be frustrated by 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. 23, in which “Sinead” answered an inquiry.


Good day Francisco, thank you for reaching out to us how may I help you today?-Sinead

hello, do you deliver to brazil?

Hi Francisco, yes we do deliver to Brazil. Here's a link for an array of items you can choose from http://bit.ly/2mwYFmB

Send Flowers and Gifts to Brazil | 1-800-Flowers.com
Send flowers to Brazil for every holiday with 1-800-Flowers.com…
1800flowers.com

thank you 🙂

Your welcome Francisco, if there's anything else I can assist you with don't hesitate to contact us as 1800flowers.com or call us as 1-800-356-9377.-Sinead

Fig. 23. Human beings sending messages thorough *1-800-Flowers.com* chatbot.

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). In fact, these 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” (Fig. 9) or its direct translation to Portuguese in “Ver na web” in the Brainstorm 9 chatbot (a Portuguese-speaking chatbot). On further analysis, both chatbots display messages stating that they were made on the same platform: Chatfuel[23]. 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 indicate, the final chatbot language, may include parts of the discourse of the three designers.

VI. Discussion

In the previous section we discussed how sign classes and strategies identified were used in a set of different chatbots, as well as how they could help designers or evaluators reflect upon their choices regarding the chatbots interactive language.

[23] https://chatfuel.com, last access on Jan/2018.

In this section, we go beyond the 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.

A. *Considerations About Chatbot Design*

Some of the main challenges 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 by including a tutorial about the features in the very first messages sent to the user (as in TechCrunch and WSJ). Other 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 by a list of main features, so the user may know what the chatbot can do and how to interact with it.

One 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 him/herself, but the chatbot would not understand that same sentence when typed by the user. That was the case with TechCrunch (Fig. 13) and WSJ, that offered a card with the title "What can I say?" in its main menu (Fig. 10), but would not understand that same sentence when typed by the user (Table III, 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 chatbots 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 "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.

Beta chatbot mainly adopted this approach, 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 scripts of conversation by offering options on how to continue most of the time, or even making it more costly to send 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 Fig. 11 and Fig. 14).

The issues discussed in this section raise the challenges that come with designing chatbots. The sign classes and strategies identified in this paper can support designers in reflecting about 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.

B. *Considerations About SIM Application to Chatbots*

As presented in the Methodology section, because 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 different system's options and menus, for they tend to be all exposed on the interface. Chatbots, on the other hand, 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 often are more paths available on a conversation than in a traditional graphical user interface because the interactions are less restraint (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 him/herself 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 III and their variations. The inspector may also define criteria of how to chose input messages to be evaluated. For instance, he/she 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 suggestions (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 on 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.

C. 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 in order 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 IV - "First Round: Results". We then proceeded to the second round of inspections, taking a top-down approach in order to consolidate the findings of the first inspections.

The top-down approach was executed by perfoming 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 class signs or strategies.

Nonetheless, it is important to point out that in the second round analysis we did find evidences of a new strategy regarding how chatbots introduce new features (see section V-A). 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.

In order 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 chatbots analyzed and then triangulate their findings to the results presented in this work.

VII. Conclusions and Future Works

Even though chatbots have been around for a long time, there are few works 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.

Our study is divided in 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.

The chatbots were inspected by two specialists, 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 6 sign classes and 11 strategies associated chatbot interactive language design.

On 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. In addition, we found evidences of a potential new strategy: Actively offering a feature to users. As mentioned on section V-B - "Using Sign Classes and Strategies to Discuss Designers' Choices", a more extensive analysis is necessary to be able to better understand this potential strategy and consolidate it as an addition to our set of strategies. Also, the sign classes were consolidated, and no evidences of any new sign classes were 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 openness of the conversational space and supporting users interaction with the chatbot, our work can be useful to both researchers interested in chatbot conversational interfaces and chatbot designers.

In order to further consolidate the strategies and sign classes discussed in this paper it would be interesting to perform new inspections on other chatbots. 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, 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 the user. However, in our analysis we found evidences that the final metacommunication is, in fact, a product written by different 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.

All of the 13 inspected chatbots used the Facebook Messenger as a 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, during our second round of inspections, we came across some communicative acts from different chatbots that seemed identical, namely in Brainstorm 9 and TechCrunch. Looking further into the issue we identified that both chatbots had been developed using the same platform (Chatfuel), as discussed in section V-B - "Using Sign Classes and Strategies to Discuss Designers' Choices". That may indicate that the chatbots' designers nay have "inherited" some of the chatbots' discourse from the development platform.

Therefore, as a future work, it could be interesting to analyze separately each one of these platforms and identify how much "say" each of these authors actually have 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 several 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 it is necessary semantic and contextual analysis of the sign in order 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 on deciding which strategies to use. Furthermore, the identified strategies 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. In addition, 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. Another interesting direction for future work would be to analyze our results and strategies in the light of linguistic theories that would allow for a better account of aspects such as semantics and pragmatics.

Acknowledgments

This work was partially supported by the project FAPEMIG-PRONEX-MASWeb (APQ-01400-14). Authors also thank CNPq, PPGCC and Google for their individual supports.

References

[1] D. G. Bobrow, "Natural Language Input for a Computer Problem Solving System," Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1964.

[2] J. Weizenbaum, "ELIZA — A Computer Program for the Study of Natural Language Communication Between Man and Machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966. [Online]. Available: http://doi.acm.org/10.1145/365153.365168

[3] H. Candello and C. Pinhanez, "Designing Conversational Interfaces," in *Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais - IHC 2016*, 15th ed. Porto Alegre, RS, Brazil: Sociedade Brasileira de Computação - SBC, 2016, vol. Parte C - Livro dos Tutoriais.

[4] M. McTear, Z. Callejas, and D. Griol, "Evaluating the Conversational Interface," in *The Conversational Interface*. Springer International Publishing, 2016, pp. 379–402. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-32967-3_17

[5] J. Jiang, A. H. Awadallah, R. Jones, U. Ozertem, I. Zitouni, R. G. Kulkarni, and O. Z. Khan, "Automatic Online Evaluation of Intelligent Assistants," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. New York, NY, USA: ACM, 2015, pp. 506–516. [Online]. Available: http://doi.acm.org/10.1145/2736277.2741669

[6] C. S. de Souza, C. F. Leitão, R. O. Prates, and E. J. da Silva, "The semiotic inspection method," in *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems*, ser. IHC '06. New York, NY, USA: ACM, 2006, pp. 148–157. [Online]. Available: http://doi.acm.org/10.1145/1298023.1298044

[7] C. S. de Souza, *The Semiotic Engineering of Human-computer Interaction*. MIT Press, 2005.

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

[9] F. A. M. Valério, T. G. Guimarães, R. O. Prates, and H. Candello, "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*, ser. IHC '17. New York, NY, USA: ACM, 2017.

[10] H. Candello, C. Pinhanez, and F. Figueiredo, "Typefaces and the Perception of Humanness in Natural Language Chatbots," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. New York, NY, USA: ACM, 2017, pp. 3476–3487. [Online]. Available: http://doi.acm.org/10.1145/3025453.3025919

[11] M. Portela and C. Granell-Canut, "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*, ser. Interacción '17. New York, NY, USA: ACM, 2017, pp. 48:1–48:7. [Online]. Available: http://doi.acm.org/10.1145/3123818.3123826

[12] C. S. Peirce, *The essential Peirce: selected philosophical writings*, N. Houser and C. J. W. Kloesel, Eds. Bloomington: Indiana University Press, 1992.

[13] N. M. Radziwill and M. C. Benton, "Evaluating Quality of Chatbots and Intelligent Conversational Agents," *arXiv:1704.04579 [cs]*, Apr. 2017. [Online]. Available: http://arxiv.org/abs/1704.04579

[14] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella, "PARADISE: A Framework for Evaluating Spoken Dialogue Agents," in *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, ser. EACL '97. Stroudsburg, PA, USA: Association for Computational Linguistics, 1997, pp. 271–280. [Online]. Available: http://dx.doi.org/10.3115/979617.979652

[15] S. de S. Reis and R. O. Prates, "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*, ser. IHC+CLIHC '11. Porto Alegre, RS, Brazil: Brazilian Computer Society, 2011, pp. 177–186. [Online]. Available: http://dl.acm.org/citation.cfm?id=2254436.2254468

[16] C. S. de Souza, C. F. Leitão, R. O. Prates, S. A. Bim, and E. J. da Silva, "Can inspection methods generate valid new knowledge in HCI? The case of semiotic inspection," *International Journal of Human-Computer Studies*, vol. 68, no. 1–2, pp. 22–40, Jan. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581909001128

[17] L. F. H. Bento, R. O. Prates, and L. Chaimowicz, "Using Semiotic Inspection Method to Evaluate a Human-Robot Interface," in *2009 Latin American Web Congress*. New York, NY, USA: IEEE, Nov. 2009, pp. 77–84.

[18] L. P. D. Corrêa, F. R. S. Coutinho, R. O. Prates, and L. Chaimowicz, "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*, ser. IHC '12. Porto Alegre, RS, Brazil: Brazilian Computer Society, 2012, pp. 47–56. [Online]. Available: http://dl.acm.org/citation.cfm?id=2393536.2393543

[19] F. A. M. Valério, T. G. Guimarães, and R. O. Prates, "Strategies for Encouraging Sharing in Social Networks for Professionals," in *Proceedings of the 15th Brazilian Symposium on Human Factors in Computer Systems*, ser. IHC '16. New York, NY, USA: ACM, 2016, pp. 54:1–54:4. [Online]. Available: http://doi.acm.org/10.1145/3033701.3033756