Do Multilingual Users Prefer Chat-bots that Code-mix? *Let's Nudge and Find Out!*

ANSHUL BAWA, Microsoft Research, India
PRANAV KHADPE, Microsoft Research, India
PRATIK JOSHI, Microsoft Research, India
KALIKA BALI, Microsoft Research, India
MONOJIT CHOUDHURY, Microsoft Research, India

Despite their pervasiveness, current text-based conversational agents (chatbots) are predominantly monolingual, while users are often multilingual. It is well-known that multilingual users mix languages while interacting with others, as well as in their interactions with computer systems (such as query formulation in text-/voice-based search interfaces and digital assistants). Linguists refer to this phenomenon as *code-mixing* or *code-switching*. Do multilingual users also prefer chatbots that can respond in a code-mixed language over those which cannot? In order to inform the design of chatbots for multilingual users, we conduct a mixed-method user-study ($N = 91$) where we examine how conversational agents, that code-mix and reciprocate the users' mixing choices over multiple conversation turns, are evaluated and perceived by bilingual users. We design a human-in-the-loop chatbot with two different code-mixing policies – (a) *always code-mix* irrespective of user behavior, and (b) *nudge* with subtle code-mixed cues and reciprocate only if the user, in turn, code-mixes. These two are contrasted with a monolingual chatbot that never code-mixed. Users are asked to interact with the bots, and provide ratings on perceived naturalness and personal preference. They are also asked open-ended questions around what they (dis)liked about the bots. Analysis of the chat logs, users' ratings, and qualitative responses reveal that multilingual users strongly prefer chatbots that can code-mix. We find that self-reported language proficiency is the strongest predictor of user preferences. Compared to the *Always code-mix* policy, *Nudging* emerges as a low-risk low-gain policy which is equally acceptable to all users. Nudging as a policy is further supported by the observation that users who rate the code-mixing bot higher typically tend to reciprocate the language mixing pattern of the bot. These findings present a first step towards developing conversational systems that are more human-like and engaging by virtue of adapting to the users' linguistic style.

CCS Concepts: • **Human-centered computing → Personal digital assistants**; **User studies**; *Empirical studies in HCI*.

Additional Key Words and Phrases: intelligent personal assistants; multilingual interfaces; code-mixing; human-agent interaction; human-centered AI

ACM Reference Format:
Anshul Bawa, Pranav Khadpe, Pratik Joshi, Kalika Bali, and Monojit Choudhury. 2020. Do Multilingual Users Prefer Chat-bots that Code-mix? *Let's Nudge and Find Out!. Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 41 (May 2020), 23 pages. https://doi.org/10.1145/3392846

Authors' addresses: Anshul Bawa, Microsoft Research, Bangalore, India, anshul.bawa.anshul@gmail.com; Pranav Khadpe, Microsoft Research, Bangalore, India, pranav.khadpe@gmail.com; Pratik Joshi, Microsoft Research, Bangalore, India, pratikmjoshi123@gmail.com; Kalika Bali, Microsoft Research, Bangalore, India, kalikab@microsoft.com; Monojit Choudhury, Microsoft Research, Bangalore, India, monojitc@microsoft.com.

2573-0142/2020/5-ART41 $15.00
https://doi.org/10.1145/3392846

1 INTRODUCTION

Developers of conversational systems are diverting an increasing amount of attention towards designing conversational agents that are more human-like. This includes incorporating and maintaining consistent personalities as is done for most commercial chatbots [19, 42, 64, 65], mimicking nuances and quirks of human communication (notably the now-famous Google Duplex "uh-hmm") [62], but more broadly, retrofitting behavioral intricacies of human-human communication to human-agent communication. However, exploring the entire space of behaviors that emerge in human conversations is prohibitively large and consequently, research, with this thrust towards anthropomorphization, focuses on iteratively incorporating more of such nuances into agent design.

A major challenge in making text-based agents more human-like is that humans are adaptive and flexible in conversations, and conversational behavior is largely dictated by the interests of people involved, their shared context, their social distance, and the cultural contexts. While humans are accustomed to adapting to people and contexts, agents need special provisions for such adaptations. Along these lines, work has looked at how responses of chatbots can be conditioned on information about individual users [35, 40], how often agents should switch topics and ask questions [57], and how more diverse responses can be generated [7]. The overarching theme, in most previous work, is that *content in human conversations is dynamic* and we need to explore how to build agents that can adapt and navigate through a range of topics, subject to the user and context. While these approaches try to capture variations in *"what"* a chatbot should say at a particular turn in the conversation, they leave unexplored the question of *"how"* a chatbot should phrase the message.

In order to move towards the goal of building human-like chatbots, we argue that it is insufficient to look at content alone. The fact that *style in human conversations is dynamic* too needs to be addressed. Human languages are rich and provide us several ways (styles) to express the same thought— we can choose to be more formal and ask, "Can you give me the directions to the restaurant, please?" or assume a more casual tone- "uh, directions?". We can choose to be more or less verbose, personal or impersonal, introduce varying amount of humour and possibly make infinitely many such choices in how we want to communicate our thoughts. Context, both interpersonal and cultural, finds its way into not just the content of conversations but also the style of utterances. Should we then design chatbots that are mindful of style and not just content, in a conversation? Would this make our chatbots more human-like?

A form of stylistic variation that has assumed special importance is that of language mixing. Speakers in multilingual societies have been known to fluently and frequently alternate between languages they are proficient in- a phenomenon known as code-mixing (CM) [30, 44]. Despite an increasing number of users of conversational systems coming from multilingual societies [2], conversational systems still remain predominantly monolingual, even as they are being developed in new languages [58]. *In such multilingual markets, does a monolingual conversational system, capable of handling interaction in one language that the user speaks, suffice?* It becomes important to understand the need for conversational systems to move beyond monolingual interactions towards such mixed language interactions.

Backed by our reasoning about the importance of style, we seek to understand if chatbots that can handle code-mixing will be perceived as more human-like by such users. This paper presents a user study with the following question:

Research Question: *Do multilingual users prefer chatbots that code-mix systematically, and if so, what is the optimal way to code-mix systematically?*

A prominent theory that seeks to explain the dynamics of style in human conversations is the Communication Accommodation Theory (CAT) [28, 29] which posits that speakers, in co-operative communication, end up reciprocating each others' styles, attempting to converge to a common style - a process called accommodation. A computational study of style-accommodation [20] shows that style-accommodation is highly prevalent and exhibits complexity in Twitter conversations. In this paper, we explore whether chatbots *should* accommodate for the CM of the individual users they interact with. We evaluate if users express a marked preference for a chatbot that reciprocates their CM over an otherwise similar chatbot that does not accommodate for their CM, the difference being only in the surface form realizations of the response. We choose to focus on users in India for pragmatic reasons including proximity and the presence of a large population of multilingual users. 10.6% of the population of India is English literate with 6.8% reporting it as their second language and 3.8% reporting it as their third language [27]. Given a smartphone penetration rate of over 25% [1], there is bound to be a sizeable population of multilingual users [36]. Thus the Indian market has been the focus of other research in multilingual interfaces as well [10, 36]. Since Hindi is the most widely spoken language in India, with English being second, we choose to work with the frequently-mixed English-Hindi (En-Hi) language pair while studying preferences of participants from India.

We devise two prototype CM policies- one that always code-mixes and another, rooted in the CAT, that does so intelligently by *nudging*- implemented as algorithms that take into account the user's CM. These CM policies run in parallel with the conversational system's response generation system and when both the response and the CM policy output are ready, a paraphrasing system introduces CM in the response in accordance to the policy's output. Using a human-in-the-loop prototype of our conversational system, we carry out a study with 91 participants from India to understand the impact of CM accommodation on user evaluations of the conversational system.

We find that participants rate bots that can code-mix higher than a monolingual English bot, in terms of human-likeness and conversational abilities. We observe that individual differences between participants are a strong predictor of their evaluations. Participants with higher fluency in the two languages, tend to reciprocate the chatbots' CM more often and evaluate the CM chatbots more favorably than participants that don't reciprocate the chatbots' CM. Similarly, when participants perceived that the chatbot was reciprocating their CM, they evaluated the chatbots' CM as more natural. Between the Always CM and the Nudge policy, we find that the Always CM policy is a high risk-high gain choice. On the other hand, the Nudge policy is a lower risk design choice with slightly lower but more consistent ratings across users.

The primary implication of our findings, for the design of chatbots that code-mix, is that in the absence of knowledge about the users' fluency and attitudes towards CM, it is better to adopt the Nudge policy while the Always CM and monolingual English systems are a good fit when it is, a priori, known whether or not the users are fluent in the constituent languages and have a positive predisposition towards CM. Our results galvanize the need to divert increasing attention towards developing language understanding and generation systems for CM language due to the unique utility these serve in a conversational setting. The accommodation policies devised in this work also extend to style dimensions beyond just CM and can be incorporated into systems that accommodate for politeness, formality and similar style choices.

2 RELATED WORK

Our work is situated in the context of previous efforts that seek to understand how humans communicate with technological systems, how language technologies have evolved, how norms from human-human communication extend to human-computer interaction, and how to develop systems that are informed by all these in the specific domain of mixed language technologies.

We give an overview of themes most relevant to our work, from literature in linguistics, human-computer interaction and natural language processing.

2.1 Computers as Social Actors

The Media Equation [52] proposes that norms from human-human communication carry over to the realm of human-machine interaction. People were found to be naturally inclined to be polite to a computer despite it's inanimacy [52]. Subsequent research has gone on to show that: people have similar processes of impression formation with robots as they do with other humans [37], derive similar psychological outcomes from disclosing to a chatbot when compared to another human [32], and even co-operate more with a computer when they are primed to think that the computer is on their "team" [48]. It is not a straightforward parallel, however - social norms around human interactions do not entirely shape expectations for human-agent conversations [17]. But largely, humans have a subconscious tendency to treat computer systems as they do other humans, and this has lead to the paradigm of "Computer as Social Actors" (CASA).

A major implication of the CASA framework, for the design of human centered systems, is that "people should be able to use what comes naturally – rules for social relationships and rules for navigating the physical world," [52] *[sic]*. People have existing notions of how to navigate the world— being polite, co-operating, resolving conflicts. Designing systems, from the lens of the CASA framework, should take into account these behavioural tendencies. For the design of conversational systems, this implies building systems that enable the user to interact with them as they would with other humans, thus justifying the need to build more human-like agents. Users in bilingual communities tend to code-mix in conversations with each other and have a natural tendency to do so in interactions with computer systems. Thus we reason that users would evaluate systems that afford such interaction, more favorably. More concretely, we hypothesize:

Hypothesis 1 (H1). *Chatbots that code-mix will be rated as more human-like and as having better conversational ability than monolingual chatbots.*

This thought has been strengthened by findings of studies that show that people tend to favor human-like conversational systems. People feel a stronger urge to resolve misunderstandings with more human-like agents [18]. Similarly, people look more favorably upon conversational agents that nod and provide envelope feedback [16]. Our work seeks to contribute to this line of work by specifically exploring how text-based agents can be made more human-like by understanding and reciprocating users style choices— specifically those concerning language mixing.

2.2 Conversational Systems and Human-likeness

As conversational agents become prevalent with designers aiming to broaden their capabilities, research in the design of conversational systems has grown from developing better language generation and language understanding models to include more human-centered considerations [34, 41, 63]. In pursuit of human-likeness, the design of conversational systems has explored the use of personalization [35, 40], personas [8, 19], visual cues including animations and embodiment [12, 15, 16], and voice to signal human traits [39, 49]. Recent work has highlighted design considerations of the "voice" of voice-based conversational systems [13] from the perspective of the users and context, including the thrust towards individualization and adapting to context. Voice-based systems and embodied systems, however, are usually built on top of a core text-based system with a layer to switch in and out of the appropriate input-output modality (a speech recognition and accompanying text-to-speech system in the case of voice-based systems). Thus, we concern ourselves with understanding what contributes to human-ness in text-based systems- at the level

of language alone- while the inquiry itself extends to other modalities such as voice-based and embodied agents.

2.3 Multilingual Conversational Systems

We are not the first to acknowledge language preferences of multilingual users. Developers of commercial systems and researchers, both, have looked at designing conversational systems for multilingual users. Systems have been developed to support interactions in multiple languages [33, 50], however, their primary focus is on enabling users to use the language they are most comfortable in while not making specific efforts to mimic human conversation or to handle code-mixing within a message. These systems require that the users' message be, entirely, in one of the supported languages and the system responds in the same language. If the user opts to change the language of the conversation, the system follows the user and responds in the new language. Our inquiry is fundamentally different in that we seek to understand how conversational systems can afford more natural and human-like interaction by appropriately accounting for users' language mixing choices. Similarly, Microsoft developed Ruuh [31]- a chatbot for the Indian market with a specific focus on human-likeness and conforming to cultural norms. Ruuh, at it's core, had a deep learning based response generator trained on code-mixed tweets from India and thus could generate code-mixed responses, however, Ruuh did not account for the language preferences of the users and would code-mix regardless of whether the user welcomed code-mixing or not. In contrast, we work within the framework of the CAT to systematize language mixing through the conversation and we concern ourselves with controlled language mixing with the goal of taking a step towards making human-agent interaction similar to human-human interaction. CAT suggests that for goals including social approval and communication efficiency, people in co-operative communication adapt to each others' style choices and attempt to reduce the differences in their communicative behaviors [28, 29]. This suggests that chatbots that adopt such strategies to adapt to and reciprocate users' style choices might appease more users as opposed to a one-fits-all solution, by virtue of adapting to the individual styles of users. We hypothesize:

Hypothesis 2 (H2). *A chatbot that adapts to users' individual styles and accommodates for their language-mixing behaviors would appease more users than a chatbot that does not accommodate for users' language-mixing tendencies.*

2.4 Code-choice as linguistic style

Code-choice is known to be associated with various sociopragmatic functions and considerations [3, 6, 11, 47, 56] that influence speakers' decisions on which code they use and when. Code-mixing (CM) can indicate a shared linguistic identity [5] and make a conversation sound more natural or engaging, convey informality, and reduce perceived social distance between speakers [14, 22, 45]. While CM is known to be similar to other dimensions of linguistic style [9, 59] in its cohesive and accommodative characteristics in human conversations, it also differs from them in being a strong sociological indicator of identity [5]. This association and dependence on sociolinguistic variables beyond the textual surface form is precisely what makes it hard for a data-driven system to capture the dynamics of this linguistic phenomenon.

Another important consideration is the wide range of differing attitudes towards CM that exist at the level of an individual or community of speakers [24] - not only sociolinguistic factors like age, gender, education and language proficiency, but also personality types of speakers (levels of emotional stability, tolerance to ambiguity, cognitive empathy and neuroticism) [23]. Thus, the preference towards a CM agent might widely vary across individuals, communities and multilingual geographies, and we cannot a priori comment on the usefulness of CM agents.

Instead of delving further into the functions and motivations for CM, we formulate our questions around its effects on a user of a chat system. Specifically, when a user interacts with two identical chatbots that differ only stylistically, specifically in the language of their messages – one always responds in English whereas the other code-mixes fluently in English and Hindi – do users systematically prefer one over the other? Does the code-mixing chatbot get consistently better or worse ratings than the monolingual chatbot?

If there is a systematic difference, we explore how much of it is due to:

- the users' demographics (their native language, proficiency in either language, ...)
- individual differences in the patterns of CM - what surface forms are considered more natural varies across speakers
- expressed attitude towards CM - not all bilingual speakers consider CM as equally grammatical or valid in written/text-based communication
- inadvertent differences in the 'persona' of the chatbot conveyed by virtue of their mixing [14, 22, 45]
- the sheer novelty of a chatbot that can converse in a mixed language
- other confounding differences in the conversations with each chatbot, like the content or topic of the conversation, response time, etc.

Since this particular form of stylistic variation has not been studied much in a human-agent conversational setting, we first explore the effects of merely introducing this style into our chatbot. We then move on to a further exploration of the effect of adding reciprocity and reciprocal variation into this style dimension.

The latter leads to a follow-up question. In a typical human-human conversation between bilinguals, the choice of code is neither premeditated nor fixed. Rather, it is known to exhibit cohesiveness and interpersonal accommodation, as shown empirically [9]. Given this knowledge, we explore if a chatbot that follows an online reciprocative CM strategy is judged differently from a chatbot whose CM is fixed and independent of whether or not the user code-mixes.

3 HUMAN-IN-THE-LOOP SYSTEM PROTOTYPE

In this section we outline the implementation of our conversational AI system prototype, which is the substrate on which we conduct our user study. Specifically, we describe our human-in-the-loop implementation of the conversational system as a whole, the design and implementation of our conversational CM policies, and the process of recruiting and training our human wizards.

3.1 Human-in-the-loop Conversational Agent

While there has been progress in the automated generation of CM text [26, 38, 51, 55, 60], current generation systems for CM text do not provide fine-grained control over the extent of mixing introduced. Similarly current language understanding modules also do not fare well on code-mixed text [51, 55]. Developing better mixed-language understanding and generation modules requires curation of mixed-language data and expensive annotation efforts. This, along with the need to develop novel computational methods for mixed-language, makes the development of code-mixing bots an expensive endeavor in terms of effort and resources. Our study seeks to understand if there is a compelling reason to undertake this endeavor. Since existing mixed-language technologies are inadequate and we don't want random errors introduced by the language generation and understanding modules to interfere with user evaluations, and so that our findings hold as technologies for CM text improve, we prototype our agent by building a human-in-the-loop paraphrasing wrapper over a monolingual chatbot. We use Mitsuku[1], a popular online chatbot, as

[1]Mitsuku can be found here: https://www.pandorabots.com/mitsuku/

our base monolingual bot and dialog manager. A human wizard serves as an intermediary with this base bot on one end, and the user on the other end. The wizard relays messages between the user and the base chatbot, and introduces and manipulates all the stylistic variation that is warranted by the experiment design. Figure 1 depicts the events that occur during a turn in the conversation and the wizard workflow. Figure 2-(a) shows the conversational interface through which the user interacts with the system and Figure 2-(b) shows the interface through which the wizards mediate the conversation. At every turn of conversation, the human wizard performs three roles in sequence:

- First, the wizard annotates the incoming user *message* with a label corresponding to the extent of CM present in accordance with the annotation schema described in 3.2.
- The wizard translates this incoming *message* from mixed language to English. The system then relays this to Mitsuku and retrieves Mitsuku's *response* to this English message.
- At this stage, one of the mixing policies (described in 3.3) informs the wizard of the amount of mixing that needs to be introduced into the *response*. The wizard paraphrases this *response* from English, to reflect the desired level of mixing. This *response* is finally sent to the user to complete the conversation turn.

3.2 Message Annotation Schema

Previous work has largely looked at binary classification of presence/absence of a style [21, 43]. Simultaneously, current language generation systems do not provide us a fine grained control over the amount of style that is introduced in the generation and only provide a binary control for presence/absence of a style [25]. However, the phenomenon of CM is full of variation and this binarization is insufficient to fully characterize it. So, we define an annotation schema to capture the varying 'extent' of code-mixing in a given text. The differences in the surface form of a piece of mixed text can be on account of a number of factors.

- The relative salience, or 'expectedness' of the two languages that are being mixed, is often different, and leads to differences in the extent of mixing [46]
- The granularity of mixing varies from the level of a conversation, a topic, a single conversation turn, to mixing within a sentence and sometimes even within a word. [61]

Specific to our experimental and usage context of bilingual users, we deem it appropriate to assume English to be the background, non-salient or *unmarked* code, and to evaluate any CM against an all-English baseline. In a text-based conversational context, we fixate on sentence-level mixing (and annotations thereof) to be the appropriate granularity to focus on.

To facilitate finer control over the measurement and generation of code-mixed responses, we come up with a simple schema to classify any given sentence based on the level or degree of mixing occurring in it. This is the schema used by the trained wizards to annotate all messages in real-time, hence the brevity in notation is designed to facilitate speed without compromising on clarity.

Specifically, the schema is based on identifying :

- The major or dominant language in the sentence - English (E) or Hindi (H). The language of the main verb(s) in the message, or the one with most number of words.
- The extent of the minor or non-dominant language in the sentence, whether it is absent (N), is present as a tag or frozen expression (T), as a lexical substitution (L) or a full phrase (P), in increasing order of mixing. For a sentence with multiple forms of mixing present, the higher level of mixing is annotated.

None (N) means that the message is entirely in the major language. "EN" denotes a message entirely in English, and "HN" a message entirely in Hindi. A sentence tag is a short addition to a sentence that often looks like a question ('The weather is nice today, *isn't it?*'). Tags and frozen

expressions are frequently borrowed words and phrases from the minor language. Insertion of tags does not indicate fluency in the minor language. Frozen or "fixed" expressions are similarly non-indicative of proficiency, and are characterized by idiomatic usage, semantic bleaching or phonological reduction. Examples include very common conversational utterances like "Thank you" and "How are you?". A person who understands Hindi but can not speak it may still be able to respond with tags like "*Accha*" and "*Theek hai*" ("Okay"). Some examples of commonly used English conversational tags and frozen expressions are - 'oh really', 'why not?', 'alright', 'exactly!', 'Isn't it?', 'good, right?', 'I don't know', 'what else?', 'I see', 'Okay then'. Some examples of commonly used Hindi conversational tags and frozen expressions are - *'kaise ho'* ('how are you'), *'bahut acha'* ('very good'), *'theek hai'* ('okay'), *'aur batao'* ('what else'), *'sahi kaha'* ('that is right'), *'haanji'* ('yes').

Lexical substitution (L) refers to replacements of individual words, often nouns and verbs, from the major language with their equivalents in the minor language. Note that we do not consider borrowed words as a lexical substitution, as their usage is less stylistically marked [4]. For example, the word 'school' is a borrowed word from English to Hindi, and can very well be considered a part of the Hindi lexicon, its usage being as common as its translation '*vidyalaya*', if not more. As a policy, we encourage wizards to, when prompted, make lexical substitutions with words that are unambiguously not borrowed.

In the case of phrasal substitution (P), entire phrases or clauses would be in the minor language. An example is "I liked the movie *par kahaani itni acchi nahi thi* (but the plot wasn't that great).".[2]

Our annotation schema is almost exclusively syntactic, and is designed to be dissociated from sociolinguistic and pragmatic considerations, though we acknowledge that the latter constitute a more holistic framework through which to analyze CM.

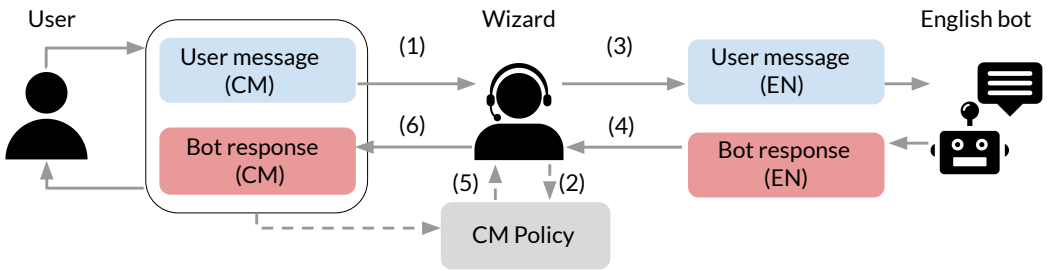

Fig. 1. A schematic representation of our human-in-the-loop conversational system prototype. A conversation turn begins with (1) the user sending a message. (2) The wizard annotates the level of code-mixing present in this message which is fed to the Policy along with the annotations of the previous turn of the chat. (3) The wizard paraphrases the user's message to English which then goes to the monolingual English chatbot at the core. (4) The bot's response to this message is then sent to the wizard along with the (5) Policy's prompt. (6) The wizard then introduces the appropriate amount of code-mixing before sending the response to the user to complete the turn of the conversation.

[2]In cases with phrasal substitution, it may be harder to identify the major language. In this example, both EP and HP seem to be plausible labels. Deciding which one suits better involves considering the language of the two clauses, the language of the verb (or verbs), and the connector between clauses (English for "I liked the movie overall", Hindi for the conjunction "par" and Hindi for "kahaani itni acchi nahi thi"). In this case, the message is more squarely HP, as Hindi seems to be the major language (the main verb, the larger clause and the connector are all in Hindi).

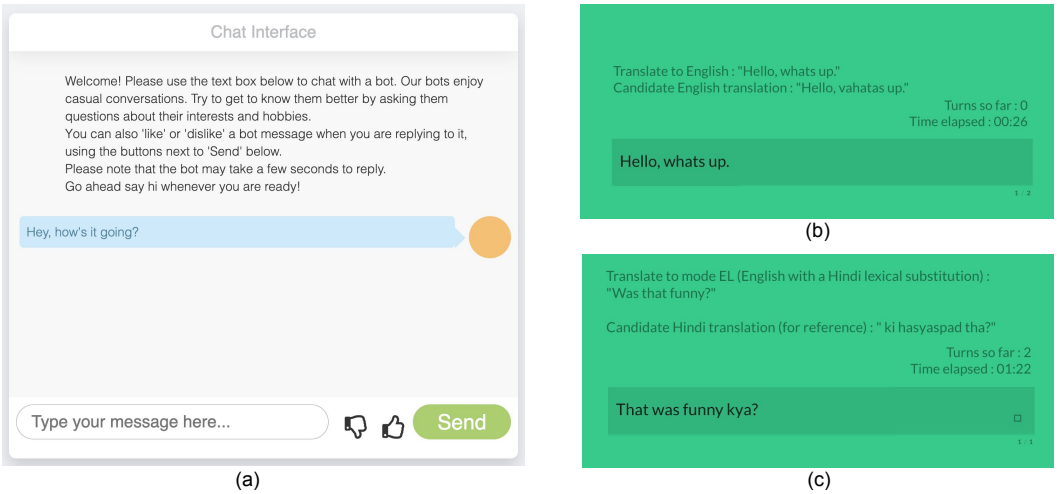

Fig. 2. (a) Our participants are exposed to a simple chat interface with a prompt to start the conversation. (b) When the participant sends in a message, the wizard annotates the level of code-mixing and is then asked to translate the message to English. The wizard is also provided with a candidate translation using the Microsoft Translator Text API. (c) When the bot reply is available, the wizard is provided with the reply as well as its Hindi translation and the prompt from the policy. Using these, the wizard adds the appropriate amount of CM in the response and sends it to the participant.

3.3 Multi-turn Code-Mixing Policies

Armed with the annotation schema that forms the vocabulary for our quantified policy formulations, we devise two different policies that dictate the language mixing that our chatbot will employ when replying to the user. The wizard is unaware of the logic that determines the extent of mixing to be added into a given message.

3.3.1 Always Mix Policy. The first policy randomly picks one of the code-mixed labels (EL, ET, EP, HL, HT, HP) at every turn. The policy never picks non-mixed labels EN or HN. Further, at every turn, the probability of a 'heavy-mixing' label (HP or EP) is the same as the probability of a label corresponding to lighter mixing (ET, EL, HT, HL). This policy does not consider the user's messages at all, and will continue to mix every turn even if the user does not mix, at all, in their replies.

3.3.2 Nudge Policy. The second code-mixing policy is slightly more complex. It is implemented as a stochastic or non-deterministic policy, to reflect the non-determinism observed in natural human code choice decisions. It is also an adaptive or online policy, in that it is sensitive to the level of code-mixing in the user's previous messages. It always starts with full English (EN), but soon introduces a small amount of Hindi into its replies (mostly ET or EL) thereby making a *nudge*.

Mathematically, we implement a nudge as taking the average mixing level of the last three conversational turns as input, increasing the level of the marked code (so EN corresponds to the lowest level of marked code and HN the highest, while all the other levels constitute a gradation between the two) by one or two levels (again, probabilistically). Changing the values of these probabilities and level jumps can make this policy more or less 'aggressive'. We arrived at the probability values used in the experiment through a small pilot and by collecting user feedback on what they *thought* the policy was doing.

Algorithm 1: Nudge policy

Result: CM level of next turn: $c0$
CM levels of last three turns: $c1, c2, c3$;
/* $c1$ and $c3$ are human turns, $c0$ and $c2$ are agent turns */
if $c1 < c2 - 1$ ***or*** $c1 + c3 < c2$ **then**
    // backoff
    $c0 = max(0, c1 - 1)$;
else
    // nudge
    $c0 = \lfloor avg(c1, c2, c3) \rfloor$;
    $r \in [0, 1, 2, 3]$ // randomness
    $c0 = min(c0 + r, 7)$;
end

Once a nudge has been made, the mixing level of the user's next message is treated as a feedback signal for the nudge. The policy could have been made to wait for two or more turns to gather feedback, but in the interest of keeping the chat conversations from becoming too long-drawn, we decided to keep the policy updating at every conversational turn. If the user's message also shows an increase in the level of mixing compared to the user's previous messages, the feedback is considered positive, the policy decides to keep the current level of mixing, and to nudge again in a few turns. If, however, the user's message does not reciprocate the change, or if the difference in the level of mixing between the messages by the two parties increases beyond a threshold, the policy does a *backoff* and restores itself to the user's level of mixing. Effectively, this policy relies on reciprocation from the user in order to continue mixing. The whole cycle of nudging and backing off repeats, and in this process, the overall level of mixing in the conversation arrives at a rate with which the user is arguably the most comfortable. The rationale behind this is that each bilingual user has an implicit preferred level of code-mixing that they are comfortable with. This includes the boundary cases of no mixing.

3.3.3 Baseline Policy - No CM. Both the Always Mix and Nudge policies above are compared against a baseline policy that always decides to reply in English (EN). However, this policy is designed to have a small probability of suggesting the wizard to add an English tag while translating or relaying messages, which we denote with a different label EET. This small modification is designed to counteract the effect of adding tags in the other two mixing policies. The mere fact that a tag is being appended can make a given sentence look more conversational, so the EET label ensures that this effect is evened out across all the variants.

3.4 Recruitment and Training of Wizards

We recruited three trained linguists (none of whom are authors of this paper) as wizards for our experiments. The wizards were provided a detailed guideline of the annotation schema and the policies. To aid the wizards in the process of paraphrasing, they were provided translations using the Microsoft Translator Text API [3] . The wizards were paid ₹250 (or roughly $4) per session, where each session took about 35 minutes of a wizard's time.[4]

[3]https://azure.microsoft.com/en-in/services/cognitive-services/translator-text-api/
[4]Wizards had multiple rounds of training and practice sessions to ensure that these guidelines were followed as uniformly as possible, and to familiarize them with the web interface to minimize potential online delays and errors later.

4 METHOD

With our human-in-the-loop conversational system in place, we conduct a user study to understand how users evaluate CM chatbots against monolingual chatbots. In this section we describe the design and execution of our user study. We first describe how we introduce two CM policies as an intervention in a pre-post design, along with measures taken to account for confounding variables. We then outline the recruitment of participants and the workflow. Subsequently, we list the measures we use in our analysis and finally highlight how we calibrated factors in our study such as response delay and inter-wizard variations.

4.1 Experimental Conditions

There are three experimental variants: one pre or 'before' treatment and two post or 'after' treatments, with the intervention being the introduction of CM by a chatbot. Each is a 15-minute chat session with a chatbot following one of the policies:

(1) **Pre** "Baseline- No CM" - The other two chatbots are compared and contrasted against this by the users while evaluating. Every user interacts with this chatbot first. This provides us a control condition, and sets the users expectations in terms of the conversational content to be expected from the chatbots.
(2) **Post 1** "Always Mix" - A bilingual En-Hi chatbot that mixes both codes at every turn. All replies by this chatbot will contain English as well as Hindi. Half of the users interact with this chatbot.
(3) **Post 2** "Nudge" - A bilingual En-Hi chatbot with the online CM policy. The other half of the users interact with this chatbot.

The experimental method is thus a hybrid of a within-subject design (Pre vs Post, with Pre acting as a control) and a between-subject design (Post 1 vs Post 2), with each user being exposed to the pre variant and exactly one of the two post variants.

We opted for a pre-post design as opposed to a control-treatment design with a randomized presentation order, because the baseline 'no CM' condition is not as much a manipulation introduced by the study as much as it is a calibration for the existing state of the world, and a proxy for the expectations of users from an existing chatbot. 80 of the 91 users had interacted with a chatbot at least once before this study. We were careful to not mention the presence of, mixing in, or proficiency in any language other than English, until the post variant is introduced and the user has finished the corresponding chat session. Exposure to a code-mixing chatbot before a monolingual chatbot, as would happen with a randomized presentation order, alters the user's expectation because of the high salience of the presence of mixed language text, and prevents a fair and unbiased evaluation of the monolingual chatbot. In other words, the novelty of the CM manipulation cannot be counterbalanced by reversing the presentation order.

The exact content of a conversation, how it unfolds, and a lot of the stylistic properties of the messages (for instance, how long or how coherent the messages are) relies on user-inputs, therefore it is not feasible to *control* for content when measuring the effects of CM. Subsequently, we adopt a study design that accounts for the effect of content by averaging it out, and keeping the content the same between the pre and post variants.

4.2 Participant Recruitment

We recruited 91 participants to sign up as users, via flyers circulated on social media. Participants were compensated with ₹500 (roughly $7) for a session lasting 35 ± 5 minutes. They had a mean age of 24.65 ($SD = 7.65$), and 66% of the participants were male. Of these, 5 participants reported using conversational agents everyday, 14 used them once a week, 10 used them twice a month, 51 rarely

used conversational agents and 11 reported never having used conversational agents. Out of the 91 participants, 46 were randomly assigned to be exposed to Pre and Post 1, while the other 45 users were assigned to Pre and Post 2. While recruiting participants, our only requirement was that they be from India and proficient in English, as the study was conducted in English. We did not impose requirements on proficiency in Hindi as teasing apart the difference between evaluations of the "one-fits-all" Always Mix policy and the "adaptive" Nudge policy was of particular interest to us. In tune with literature, we reasoned that proficiency in Hindi would affect evaluations of the two systems [10]. Having recruited participants from India, we expected participants to have varying degrees of proficiency in Hindi. In order to do a post-hoc analysis of differences in the baseline condition of Hindi proficiency, we asked participants to self-report their Hindi proficiency levels. We observed that the distribution of participants in both groups, in terms of Hindi proficiency, was largely similar ($\chi^2(2, 91) = 0.188, p = 0.91$) . In the group that was assigned Pre and Post-1, 18.18% reported having basic proficiency, 31.82% reported having near native proficiency, and 50% reported having native proficiency. In the Group that was assigned Pre and Post-2 15.55% reported having basic proficiency, 35.57% reported having near native proficiency, and 48.88% participants reported having native proficiency.

4.3 Study Workflow

Participants are first asked demographic questions, about their proficiency in Hindi and English, their educational background, cities they've lived in, and their prior exposure to chatbots. Following this, participants interact with the English monolingual chatbot. This lasts for either 15 conversational turns or 15 minutes, whichever is satisfied earlier. They are asked for feedback on this interaction. Next, the participants interact with one of the two CM chatbots. Again, this interaction lasts for 15 turns or 15 minutes. Participants then conclude the study providing feedback on the second interaction, and comparative feedback between the two.

4.4 Measures

In order to capture user preference, we use multiple kinds of sources of information for feedback, that address both stated and revealed preferences:

Individual conversation feedback: To capture user evaluations at the level of the conversation as a whole, we ask them to evaluate each bot they interact with on its *Ability to talk like a human (Human-ness)* and its *Conversational skills*, each on a 7-point Likert scale. We also ask the following subjective questions: *1) What did you like and dislike about this conversation? 2) Any specific remarks on the bot and its dialog?* Additionally, after collecting these judgments and remarks, we ask the following questions regarding the second bot: *1) Did you notice that the second bot used Hindi? 2) Did the bot's use of mixed language feel forced or natural? 3) Did you feel that the language of the bot's replies was reciprocative of your own choice of language (English, Hindi or Hinglish)?*

Comparative feedback: We include questions explicitly asking to compare the control monolingual chatbot and the CM chatbot they interacted with. These questions are: *1) Among the two bots you interacted with, which one would you prefer to continue interacting with? 2) Did you notice any differences between the two chatbots? If yes, please briefly describe them.*

Chatlog measures: We derive two types of data from the chat logs, a boolean variable indicating whether or not the user code-mixes, and the 'accommodation score' of the user. For each dialog, from both the user and the bot, the wizard either annotates or receives a label indicating the level of Hindi in the response. We represent the 8 labels numerically from 0 to 7 (0 for EN, 7 for HN), and treat it as a numeric feature for measuring the level of Hindi, which is the marked code. We

calculate the accommodation score as:

$$AccoScore(C) = \frac{\sum_{n=1}^{d} |D^{C}_{user}(i) - D^{C}_{bot}(i)|}{d} \quad (1)$$

where C is the chat, d is the number of user-bot dialog pairs, and D^{C}_{user} and D^{C}_{bot} refer to the level of Hindi used by the user and bot respectively. *The lower the score, the more the accommodation between the user and the chatbot.*

Annotated subjective feedback In order to extract better insights from the subjective feedback, we manually annotate participant feedback, with a set of categorical features:

- Naturalness: If the user felt the CM from the chatbot was natural (Forced/Natural)
- Reciprocative: If the user felt that the bot reciprocated their code (Yes/No/Unsure),
- Slow: If the user felt that the bot was slow (Yes/No/None),
- Irrelevant: If the user felt the bot's responses were irrelevant or out-of-context (Yes/No),
- Effect on Judgement: Whether the bot's CM affected the participant's evaluation positively, negatively or neither (Positive/Negative/Neutral).

4.5 Calibration through Pilots

4.5.1 Controlling for response delay. The pipeline for all the three variants has a human in-the-loop, but the complexity of work that a wizard performs per turn varies between the mixing and baseline policies. The average response time is thus prone to vary between the three variants. However, since we are interested in investigating if there is utility in chatbots code-mixing and if there is merit in directing efforts towards development of fully-automated code-mixing chatbots, we want to isolate out the effect of the various language mixing policies on user evaluations. Thus response time becomes a confounding factor to be accounted for. During our pilots, we recorded the average response times of the wizards in the CM policy settings, and used the larger of the two (the Nudge policy is slower to implement in real-time) as a minimum delay to be introduced. This approximate threshold delay (20 seconds) is then applied to each turn in the faster variants, so even if the system is ready with a response before this minimum threshold, the system waits for the remainder of the time before spitting out a response. While response time does affect user experience, we want to be able to take a peak into a future where all these variants are automated, in order to study our research question— to determine if such a future is worth developing by investing resources and efforts. So, we want to minimize the effect of response time, and only ascertain the merits/demerits of the language mixing itself. Thresholding the delay provides us a glimpse into that future at the cost of sub-optimal user experience which is uniform across the variants being studied. We further discuss the implications of thresholding the delay in the discussion.

4.5.2 Inter-wizard differences. Since wizards are an important part of the experiment setup, we had practice sessions to train the wizards and to try and minimize the differences in ratings across wizards. Wizards were trained to use the interface and were provided with instructions and example paraphrases corresponding to the different levels of code-mixing that the policies work with. Since the response itself was generated by the English bot and the amount of code-mixing to be introduced was determined by the policy, wizards had minimal leeway in the workflow. The pilots were conducted with four human wizards, and three of the wizards who consistently followed the guidelines were selected for the full study, none of whom were the authors of this paper. Through our pilots we observed no significant differences in the ratings of different wizards for a given bot variant.

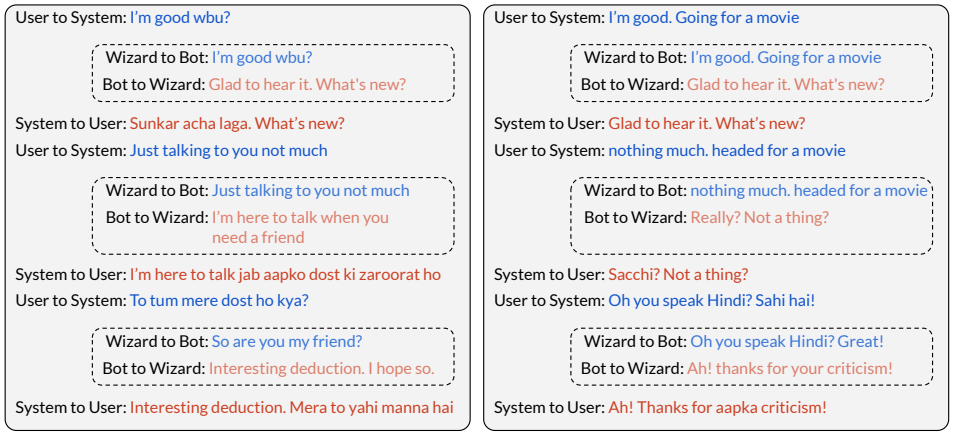

Fig. 3. Sample conversations: (left) conversation between a participant and the Always CM policy, (right) conversation between a participant and the Nudge policy.

5 RESULTS

We seek to understand the difference in ratings of the monolingual bot and the CM bots as well as the difference in the ratings of the two policies. We also aim to understand what factors might drive this difference in rating, other than the differences in design alone.

5.1 Monolingual vs. CM bots

Participants rated the CM chatbots higher than the monolingual bots on both human-likeness and conversational ability. We clubbed together the ratings of the Always CM chatbot and the Nudge chatbot to obtain a cumulative of 91 ratings for the CM chatbots. Comparing these ratings of the CM chatbots, taken together, with the ratings of the monolingual chatbot, we find that participants favored both the conversational ability (Average conversation rating: $5.28 > 4.46$) ($t(91) = 4.72$, $p < 0.001$) as well as the human-ness (Average human-ness rating: $5.24 > 4.39$) ($t(91) = 4.49$, $p < 0.001$) of the CM bots. Figure 4-(a) shows that average ratings for the bilingual bots are higher than the monolingual bot. These results support H1. Our qualitative analysis (Section 5.7) sheds light on some reasons why the participants might have evaluated the CM bots more positively.

5.2 Effect of language proficiency

More proficient bilingual speakers were better able to discern how well the bot is able to integrate CM in the conversation as compared to a basic bilingual user. Taking the ratings of both CM bot variants together, we analyzed the evaluations when separated out based on participants' Hindi proficiency. Although the average ratings in each proficiency class are similar, there is a stronger agreement (less variance) over the conversational ability of the chatbot for more proficient bilingual users. As observable in Figure 4-(b), the variance of conversational ratings was much lower for those who have native proficiency, whereas the ratings varied more for less proficient user classes.

5.3 Effect of Mixing by User

Users who themselves code-mixed in conversation with the bilingual chatbots rated them higher than those who didn't. Presumably, this engagement reflects the more enthusiastic nature of users towards such chatbots. We classified users into two groups: those who also code-mixed during conversation with the bot and those who did not. Figure 5-(a) shows the average human-ness

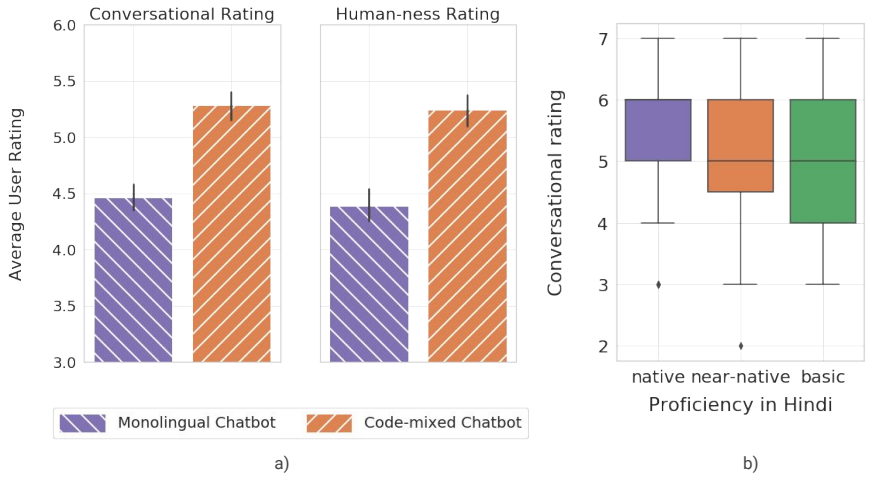

Fig. 4. a) Average ratings of users are higher for both conversational and human-ness ratings, showing that users prefer code-mixing chatbots over monolingual chatbots. b) Proficient bilingual users are better able to assess code-mixing ability of bot, having higher agreement and less variance in ratings, compared to those with less proficiency.

ratings of these users, categorized by whether they mixed or not. On average, the users who mixed have rated the bot as more human-like compared to those that did not ($t(89) = 2.17$, $p < 0.05$). To explain for this difference, we frame three user classes which can focus our further analyses:

- *User notices the bot's CM, but doesn't respond*: The user may not respond with CM due to various factors such as lack of proficiency or lack of enthusiasm towards mixing.
- *User doesn't notice the bot's CM, and hence doesn't respond*: This could be the result of the bot's CM policy being too weak, or due to the subtlety of the mixing (such as a simple frozen expression). We reason that these users will rate the chatbot on other confounding factors not pertaining to the bilingual nature of the bot.
- *User notices the CM, responds with CM*: The user is probably attempting to engage with the bot, and is reacting positively to the CM. This user group better informs us about the quality of the dynamics of bilingual conversation with the bot.

Selecting only the group of users who code-mixed in the conversation (64 out of 91 participants), we observe that although the average human-ness ratings for the Nudge bot were lower compared to the Always Mix bot ($5.38 < 5.5$), the standard deviation was lower for the Nudge bot ($1.071 < 1.198$), implying more agreement in ratings. Thus we find weak evidence for H2 when we isolate out users that engaged in code-mixed themselves.

5.4 How perceptions of linguistic reciprocation impact perceptions of naturalness

When the participants felt that the CM chatbot was reciprocating their CM, they rated the bot as more natural. We analyzed, based on user feedback, whether they felt that the chatbot appropriately reciprocated the CM level and style of the user. Figure 5-(b) shows the normalized count (normalized by clubbing together those who thought CM bot was forced and those who thought it was natural) of users categorized by whether they thought the chatbot reciprocated their language use, and split by whether the users felt the CM was forced or natural. A larger proportion of people who felt the bot was natural also felt that it reciprocated effectively, and similarly a larger proportion of people

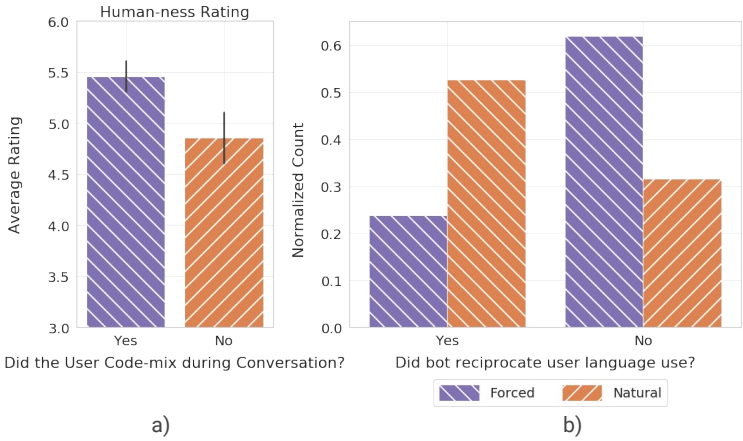

Fig. 5. a) Users who code-mixed during interaction with the CM chatbot rate it as more human-like. b) Better style reciprocation by the chatbot makes users perceive the conversation as more natural.

who felt the bot was forced also felt that it didn't reciprocate appropriately. It can be inferred from the above results that effective reciprocation plays an important role in making the conversation natural. This further motivates the need to design effective CM policies which reciprocate the user's style, such as the Nudge policy.

5.5 Nudge policy vs. Always Mix policy

Figure 3 shows examples of conversations in the two conditions. In order to understand the fine-grained effects that our policies had on our users, we compared and contrasted the two. Those users who didn't code-mix themselves, might not have noticed the bot mixing, or weren't enthusiastic or active in engaging with the code-mixing of the bot. Therefore, in order to eliminate the confounding factors, we focused only on the users who themselves code-mixed.

One specific trend of interest is the variation in the accommodation scores for different policies, as shown in Figure 6-(a). There is less accommodation on average for users in both classes for the Always Mix bot when compared to the Nudge bot, with the difference being more stark in users who don't CM. From the above observations, we conclude that the Nudge bot and Always Mix bot handle the risk-gain trade-off of code-mixing differently (risk of displeasing the user by code-mixing, gain of satisfaction of the user). The Nudge bot is low-risk low-reward, in that it subtly introduces CM without being too invasive in handling the linguistic dynamics of the conversation. However, this subtlety yields low reward, in terms of the noticeability of the mixing, and also because these interactions aren't long enough to capture later rewards. The Always Mix bot is high-risk high-reward, which means it makes it obvious that it is code-mixing, and is more in-the-face. This can result in high reward if the users prefer a high level of CM, but it can be risky when it is used by users who don't code-mix or don't react positively to it.

5.6 Qualitative Analysis

Upon looking at responses to the subjective questions which categorized how the participants felt about the chatbots' CM. 53% of the users expressed that the presence of CM influenced their ratings to be more positive. Reasons for this ranged from 'feeling more comfortable', 'better conversational flow', 'more relatable and realistic', and 'bot was friendlier'. These were precisely the sentiments that we hoped the introduction of CM would imbue in the user. Users also expressed the fact that a

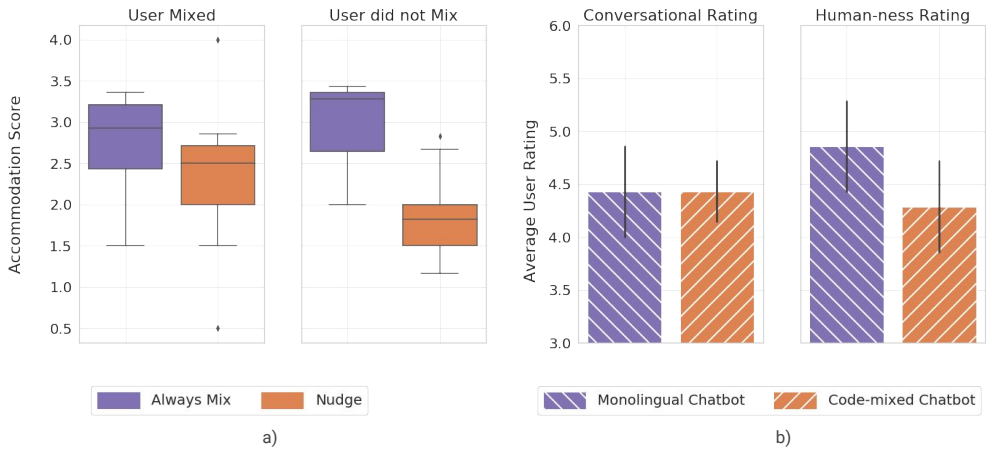

Fig. 6. a) The nudge bot accommodates better to the user's style choice compared to the Always Mix bot whether user mixes or not, more apparently when they do not mix. b) Content has an independent effect. When we isolate out participants that reported that the CM bots were contextually incoherent, we observe that for these participants the ratings for human-ness of the monolingual bot are higher than the CM bots while the conversational ability ratings are at par.

bilingual chatbot would be favorable given their natural tendency to code-mix in conversation with other bilinguals, giving examples of their parents, friends, and relatives (e.g. "I liked the second bot more. Generally, I use both Hindi and English while conversing with my friends or family."). On the other hand, only 13% of the participants felt that CM negatively affected their judgement. Some reasons cited for this were- 'informal', 'awkwardly constructed sentences', and 'too much mixing'. These reasons stress the fact that CM is a user-specific choice, and puts emphasis on the creation of smarter CM policies which appropriately gauge the language choice of the user. The remaining 34% expressed that CM did not affect their evaluations suggesting that content still plays a major role in user satisfaction.

6 DISCUSSION

We now discuss the implications of our findings and how our observations can inform the design of chatbots. We illustrate how our policies themselves can be incorporated into chatbots and how they extend to style dimensions beyond CM. Finally, we discuss the challenges faced in the study, the limitations of our study, and the implications of that for the design of studies that seek to untangle content and style and understand their individual effects on user evaluations.

6.1 Implications for the design of conversational systems

Based on our observations, we reason about how to design conversational-systems for users, taking into account their style-preferences in terms of code-mixing.

6.1.1 Chatbots should code-mix. Our findings indicate that multilingual users show a marked preference for bots that can code-mix over a monolingual English bot evidencing the merits of realizing chatbots that code-mix. This suggests the need to divert more attention towards building more robust CM language understanding systems to enable such interactions as well as better language generation systems for CM text that provide control over the extent of CM.

6.1.2 Nudge when uncertain about users' preferences. We observe that the baseline policy- No CM is a "no risk-no gain" design decision in that it's the norm. Meanwhile, the Always CM policy is a "high risk-high gain" choice. When users have a positive attitude towards CM they rate it well but when they are not enthusiastic about code-mixing, they are understandably more critical of it. In contrast, the Nudge policy receives slightly lower but more consistent ratings implying that it is a "low risk-low gain" choice. It so happens in our sample that the users that have a positive attitude towards CM far outweighed those that didn't, hence skewing the ratings of the Always CS policy, but this might be a more favourable and less representative case. Understanding the complete distribution of users' preferences, a priori, is often infeasible. In the unlikely scenario that we do know our users' preferences, it makes sense to stick to either the No CM or Always CM policy depending on whether the user is enthusiastic about CM or not. Otherwise, it is advisable to Nudge. Alternately, it is also possible to Nudge for first few turns to accurately estimate or verify the user's attitude towards CM and then move to a static policy and settle at a particular level of CM that the user is comfortable with.

6.1.3 Nudge when uncertain about users' fluency in the languages. We find that the predominant factor indicating preference for CM chatbots is fluency in both languages. Fluent users tend to reciprocate CM more often, however non-fluent users don't. This is an ideal setting for the Nudging policy as it can actively ramp up the amount of CM based on reciprocation. For fluent users that CM more, the Nudge policy would learn to increase CM tending towards the Always CM policy. In the case of users that don't reciprocate due to a lack of fluency, the Nudge bot can minimise user dissatisfaction. Our study took place in a multilingual setting where users were expected to be fluent and therefore the Always CS policy might prove more useful in such a setting. However, fluency of all users is rarely known beforehand and the Nudge policy, being a conservative choice, would be the recommended policy.

6.2 Generalizability of our policy design

We now discuss how the CM accommodation policies devised by us can be transferred to conversational systems and how they extend beyond just CM as a style dimension.

6.2.1 Incorporating CM policies into mixed language conversational agents. The main technological limitation that prevents the transfer of our policies to existing systems is the absence of natural language systems that can understand the functional role of the presence and extent of CM. Simultaneously, current CM language generation systems lack fine grained control over the extent of CS at the level of our schema and while they try to seek 'naturalness', they fall short of the mark [51, 55]. In the absence of these two critical components, transferring the policies to chatbots still requires a human-in-the-loop approach. However, as the underlying technology stack improves, our policies in their current form could serve to introduce systematic CM in chatbots.

6.2.2 Extending policies to other style dimensions. Although our paper specifically details our accommodation policies in the context of CM, the policies themselves are agnostic to the style of interest. The policy requires one subsystem that can accurately measure the 'amount' of a particular style in a user response and a second subsystem that can introduce the desired 'amount' of the style into a given bot response. Thus, we reason that these policies immediately extend to any style dimension for which such subsystems can be built. These style dimensions could include formality, accents, registers and other such shallow linguistic variations. These policies are not unique to the domain of conversational systems and are akin to work that has looked at how to model human-states in the context of semi-autonomous vehicles [54]. Future work can draw insights from policies studied in other contexts to come up with more robust policies for conversational systems.

This work also hopes to provide impetus to development of models that explicitly reason about linguistic style and do not merely rely on style implicitly emerging from data.

6.3 Challenges and Limitations

We identify two main limitations of this work. The first is a practical consideration— given the nature of our human-in-the-loop conversation system, users are exposed to non-negligible latency and the conversations are subject to random errors of the underlying conversation engine, both of which are non-ideal for our study. The second limitation lies in our formulation that assumes that content and style can be disentangled. We discuss the extent to which this holds and suggest design considerations for future studies. We also shed light on the generalizability of our findings to other multilingual populations.

6.3.1 Limitations of system design. It was observed that roughly 15% of the users commented that the chatbot was slow in responding, which turned out to be an unavoidable repercussion of the human-in-the-loop setup and the added delay to make the response times uniform. Developing automated code-mixing systems requires creation of mixed-language resources and developing mixed language understanding and generation systems. Both of these together pose a herculean task. There has been very little incentive to pursue this line of work as there was no evidence, so far, that this might enhance experiences for multilingual users. The motivation to develop such systems hinges on evidence of their utility, however, the search for this evidence is deterred by the very need to develop these systems first in order to test their utility. In light of this deadlock, our study, while imperfect, shows promise in pursuing this direction and further motivates the need to invest efforts and resources towards developing automated code-mixing systems. As future work, we hope to redo the study with automated versions of our chatbot variants and test whether our results are sensitive to response times.

Around 20% of users mentioned that the chatbot had issues with the relevance of its responses (couldn't hold the conversation effectively, wasn't understanding the context of the user dialogues, or was making irrelevant dialogues). To delineate the importance of content, we isolated out participants that complained of poor contextual relevance in the case of the CM bots but not in case of the monolingual bot. For these 7 participants, we observe that the trend is the exact opposite with the monolingual bot performing better ($M = 4.85, SD = 1.34$) than the CM bots ($M = 4.28, SD = 1.254$) in terms of human-ness ratings and the monolingual ($M = 4.43, SD = 1.27$) and CM bots ($M = 4.43, SD = 0.97$) performing at par in case of conversational ability as shown in (Figure 6-(b)). This suggests that content might have an independent effect and both need to be studied to completely characterize the dynamics in the conversation.

6.3.2 Limitations of formulation. Our formulation of the policies assumed that content and style were independent dimensions and one could be varied without variation in the other. However, as we noticed on further investigation, the degree to which content and style can be untangled depends on the choice of the style dimension, and the task itself. For example, if we choose verbosity to be a style dimension, we can say that "k" and "It's okay" convey the same content but if we choose sentiment to be a style parameter, we're unable to attribute whether the difference between "this place is great" and "this place isn't great" is that of content or style. In the case of CM, while the two can be dissociated to a large extent, there are several situations where this does not hold. CM conveys multiple sociopragmatic functions, and not all content can be expressed in mixed language without compromising on clarity. When talking about technical topics speakers are less prone to CM whereas greetings, opinions, emotional expressions are likely to contain CM. This suggests that content and style are not entirely separable Therefore, future work could try to address both content and style together.

6.3.3 Design of studies that seek to untangle content and style. Studying style independent of content poses a unique challenge. One possible solution is to exercise complete control of content by showing the users manually curated conversations that have near similar content and vary only in the particular style, but stated preferences of users may vary from actual preferences. Users might have different opinions about whether a chatbot should vary style when they observe a conversation as opposed to when they interact. A second possibility is to scale to very large sample size, where the effect of content will eventually cancel out, but a study at this scale is not feasible with the current human-in-the-loop approach. Future work could explore a large scale study with a fully automated system by choosing an appropriate style for which automatic measurement and generation techniques exist. A third approach, which serves as a middle ground, is conducting small to medium scale studies with deep qualitative interviews. This also enables researchers to explore new questions that emerge- on the fly- but it suffers in that it is myopic with respect to user distributions.

6.3.4 Generalizability of our findings to other multilingual populations. We chose to study users from India and we caution against generalizing our findings to other multilingual populations. It has been observed that even within bilingual communities that speak the same two languages- such as the English-Spanish speaking communities of San Diego and Barcelona- users have different language mixing behaviors depending on their first language- English in the case of San Diego and Spanish in the case of Barcelona [53]. Thus we might observe different outcomes if a similar study were to be conducted with English-Hindi bilingual users outside of India. However, this further motivates the need for an adaptive policy akin to our Nudge policy that can infer users' preferences and adapt accordingly, irrespective of the sociolinguistic dynamics specific to the mixing pair.

7 CONCLUSION

Our key findings are that systems that explicitly reason about the extent of code-mixing in their replies, independent of content, fare better in terms of user evaluations with bilingual users. Through a mixed-method user study, we comprehensively analyze the effects of user demographics, expressed attitudes, and language proficiency on evaluations of CM chatbots. Our contributions include a new syntactic schema for sentence-level code-mixing, and an online policy formulation to gradually incorporate any graded style dimension into a conversation based on mutual reciprocation. We also emphasize the utility of adaptively estimating hidden user preferences in cases where they are not known a priori, using insights from work on Communication Accommodation Theory. Finally, demonstrate the utility of incorporating linguistic style into dialog planning, through the proposed strategy of 'nudging'. We thus present a step towards developing more responsive, natural and human-like conversational systems that move beyond just considerations of content and take into account the linguistic style choices of users.

ACKNOWLEDGMENTS

We would like to thank Sakshi Kalra, Yash Sinha and Niyati Bafna for their contributions to this work, and all the participants in our pilots and study for their time and feedback. We would also like to thank Steve Worswick for allowing us access to Mitsuku for the purpose of this study.

REFERENCES

[1] 2019. Top Countries by Smartphone Users. (2019). https://newzoo.com/insights/rankings/top-countries-by-smartphone-penetration-and-users/

[2] Ana Inés Ansaldo, Karine Marcotte, Lilian Scherer, and Gaelle Raboyeau. 2008. Language therapy and bilingual aphasia: Clinical implications of psycholinguistic and neuroimaging research. *Journal of Neurolinguistics* 21, 6 (2008), 539–557.

[3] Peter Auer. 1995. The pragmatics of code-switching: A sequential approach. *One speaker, two languages: Cross-disciplinary perspectives on code-switching* (1995), 115–135.

[4] Peter Auer. 1999. From codeswitching via language mixing to fused lects: Toward a dynamic typology of bilingual speech. *International journal of bilingualism* 3, 4 (1999), 309–332.

[5] Peter Auer. 2005. A postscript: Code-switching and social identity. *Journal of pragmatics* 37, 3 (2005), 403–410.

[6] Peter Auer. 2013. *Code-switching in conversation: Language, interaction and identity*. Routledge.

[7] Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. 2018. Generating More Interesting Responses in Neural Conversation Models with Distributional Constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3970–3980. https://doi.org/10.18653/v1/D18-1431

[8] Gene Ball, Dan Ling, David Kurlander, John Miller, David Pugh, Tim Skelly, Andy Stankosky, David Thiel, Maarten Van Dantzich, and Trace Wax. 1997. Lifelike computer characters: The persona project at Microsoft research. *Software agents* (1997), 191–222.

[9] Anshul Bawa, Monojit Choudhury, and Kalika Bali. 2018. Accommodation of Conversational Code Choice. *Third Workshop on Computational Approaches to Linguistic Code-switching, ACL 2018* (2018).

[10] Anshul Bawa, Monojit Choudhury, and Kalika Bali. 2018. User Perception of Code-Switching Dialog Systems. In *15th International Conference on Natural Language Processing*. 171.

[11] Rafiya Begum, Kalika Bali, Monojit Choudhury, Koustav Rudra, and Niloy Ganguly. 2016. Functions of Code-Switching in Tweets: An Annotation Framework and Some Initial Experiments. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association (ELRA).

[12] Linda Bell and Joakim Gustafson. 1999. Interaction with an animated agent in a spoken dialogue system. In *Sixth European Conference on Speech Communication and Technology*.

[13] JULIA CAMBRE and CHINMAY KULKARNI. 2019. One Voice Fits All? Social Implications and Research Challenges of Designing Voices for Smart Devices. (2019).

[14] Antoinette Camilleri. 1996. Language values and identities: Code switching in secondary classrooms in Malta. *Linguistics and education* 8, 1 (1996), 85–103.

[15] Justine Cassell, Timothy Bickmore, Mark Billinghurst, Lee Campbell, Kenny Chang, Hannes Vilhjálmsson, and Hao Yan. 1999. Embodiment in conversational interfaces: Rea. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 520–527.

[16] Justine Cassell and Kristinn R Thorisson. 1999. The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence* 13, 4-5 (1999), 519–538.

[17] Leon Ciechanowski, Aleksandra Przegalinska, Mikolaj Magnuski, and Peter Gloor. 2018. In the shades of the uncanny valley: An experimental study of human–chatbot interaction. *Future Generation Computer Systems* (2018).

[18] Kevin Corti and Alex Gillespie. 2016. Co-constructing intersubjectivity with artificial conversational agents: people are more likely to initiate repairs of misunderstandings with agents represented as human. *Computers in Human Behavior* 58 (2016), 431–442.

[19] Sonam Damani, Nitya Raviprakash, Umang Gupta, Ankush Chatterjee, Meghana Joshi, Khyatti Gupta, Kedhar Nath Narahari, Puneet Agrawal, Manoj Kumar Chinnakotla, Sneha Magapu, et al. 2018. Ruuh: A Deep Learning Based Conversational Social Agent. *arXiv preprint arXiv:1810.12097* (2018).

[20] Cristian Danescu-Niculescu-Mizil, Michael Gamon, and Susan Dumais. 2011. Mark my words!: linguistic style accommodation in social media. In *Proceedings of the 20th international conference on World wide web*. ACM, 745–754.

[21] Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, 250–259.

[22] Anna De Fina. 2007. Code-switching and the construction of ethnic identity in a community of practice. *Language in society* 36, 3 (2007), 371–392.

[23] Jean-Marc Dewaele and Li Wei. 2013. Is multilingualism linked to a higher tolerance of ambiguity? *Bilingualism: Language and Cognition* 16, 1 (2013), 231–240.

[24] Jean-Marc Dewaele and Li Wei. 2014. Attitudes towards code-switching among adult mono-and multilingual language users. *Journal of Multilingual and Multicultural Development* 35, 3 (2014), 235–251.

[25] Jessica Ficler and Yoav Goldberg. 2017. Controlling Linguistic Style Aspects in Neural Language Generation. *CoRR* abs/1707.02633 (2017). arXiv:1707.02633 http://arxiv.org/abs/1707.02633

[26] Saurabh Garg, Tanmay Parekh, and Preethi Jyothi. 2018. Code-switched Language Models Using Dual RNNs and Same-Source Pretraining. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Association for Computational Linguistics, Brussels, Belgium, 3078–3083. https://doi.org/10.18653/v1/D18-1346
[27] Registrar General and Census Commissioner India. 2011. Census of India 2011. (2011).
[28] Howard Giles. 2007. *Communication accommodation theory*. Wiley Online Library.
[29] Howard Giles, Donald M Taylor, and Richard Bourhis. 1973. Towards a theory of interpersonal accommodation through language: Some Canadian data. *Language in society* 2, 2 (1973), 177–192.
[30] John J Gumperz. 1982. *Discourse strategies*. Vol. 1. Cambridge University Press.
[31] Khyatti Gupta, Meghana Joshi, Ankush Chatterjee, Sonam Damani, Kedhar Nath Narahari, and Puneet Agrawal. 2019. Insights from Building an Open-Ended Conversational Agent. In *Proceedings of the First Workshop on NLP for Conversational AI*. 106–112.
[32] Annabell Ho, Jeff Hancock, and Adam S Miner. 2018. Psychological, relational, and emotional effects of self-disclosure after conversations with a chatbot. *Journal of Communication* 68, 4 (2018), 712–733.
[33] Netiks International. 2019. Genetiks. (2019). https://genetiks.net/
[34] Mohit Jain, Pratyush Kumar, Ramachandra Kota, and Shwetak N. Patel. 2018. Evaluating and Informing the Design of Chatbots. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. ACM, New York, NY, USA, 895–906. https://doi.org/10.1145/3196709.3196735
[35] Chaitanya K. Joshi, Fei Mi, and Boi Faltings. 2017. Personalization in Goal-Oriented Dialog. *CoRR* abs/1706.07503 (2017). arXiv:1706.07503 http://arxiv.org/abs/1706.07503
[36] Naveena Karusala, Aditya Vishwanath, Aditya Vashistha, Sunita Kumar, and Neha Kumar. 2018. " Only if you use English you will get to more things" Using Smartphones to Navigate Multilingualism. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–14.
[37] Nicole C Krämer, Astrid von der Pütten, and Sabrina Eimler. 2012. Human-agent and human-robot interaction theory: Similarities to and differences from human-human interaction. In *Human-computer interaction: The agency perspective*. Springer, 215–240.
[38] Ying Li and Pascale Fung. 2012. Code-Switch Language Model with Inversion Constraints for Mixed Language Speech Recognition. In *Proceedings of COLING 2012*. The COLING 2012 Organizing Committee, Mumbai, India, 1671–1680. https://www.aclweb.org/anthology/C12-1102
[39] Max M Louwerse, Arthur C Graesser, Shulan Lu, and Heather H Mitchell. 2005. Social cues in animated conversational agents. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition* 19, 6 (2005), 693–704.
[40] JM Lucas, F Fernández, J Salazar, J Ferreiros, and R San Segundo. 2009. Managing speaker identity and user profiles in a spoken dialogue system. *Procesamiento del lenguaje natural* 43 (2009), 77–84.
[41] Ewa Luger and Abigail Sellen. 2016. "Like Having a Really Bad PA": The Gulf Between User Expectation and Experience of Conversational Agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 5286–5297. https://doi.org/10.1145/2858036.2858288
[42] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing Dialogue Agents via Meta-Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5454–5459. https://doi.org/10.18653/v1/P19-1542
[43] Rada Mihalcea and Stephen Pulman. 2007. Characterizing humour: An exploration of features in humorous texts. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 337–347.
[44] Carol Myers-Scotton. 1993. *Dueling Languages: Grammatical Structure in Code-Switching*. Claredon, Oxford.
[45] Carol Myers-Scotton. 1995. *Social motivations for codeswitching: Evidence from Africa*. Oxford University Press.
[46] Carol Myers-Scotton. 2005. *Multiple voices: An introduction to bilingualism*. Wiley-Blackwell.
[47] Mark Myslín and Roger Levy. 2015. Code-switching and predictability of meaning in discourse. *Language* 91, 4 (2015), 871–905.
[48] Clifford Nass, BJ Fogg, and Youngme Moon. 1996. Can computers be teammates? *International Journal of Human-Computer Studies* 45, 6 (1996), 669–678.
[49] Clifford Ivar Nass and Scott Brave. 2005. *Wired for speech: How voice activates and advances the human-computer relationship*. MIT press Cambridge, MA.
[50] Ocelot. 2019. Ocelot Bilingual Chatbot. (2019). https://www.ocelotbot.com/news/bilingual-chatbot/
[51] Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1543–1553.
[52] Byron Reeves and Clifford Ivar Nass. 1996. *The media equation: How people treat computers, television, and new media like real people and places*. Cambridge university press.
[53] Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1971–1982.

[54] Dorsa Sadigh, S Shankar Sastry, Sanjit A Seshia, and Anca Dragan. 2016. Information gathering actions over human internal state. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 66–73.
[55] Bidisha Samanta, Sharmila Reddy, Hussain Jagirdar, Niloy Ganguly, and Soumen Chakrabarti. 2019. A Deep Generative Model for Code-Switched Text. *arXiv preprint arXiv:1906.08972* (2019).
[56] Carol Myers Scotton and William Ury. 1977. Bilingual strategies: The social functions of code-switching. *International Journal of the sociology of language* 1977, 13 (1977), 5–20.
[57] Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? How controllable attributes affect human judgments. *CoRR* abs/1902.08654 (2019). arXiv:1902.08654 http://arxiv.org/abs/1902.08654
[58] Heung-Yeung Shum, Xiaodong He, and Di Li. 2018. From Eliza to XiaoIce: Challenges and Opportunities with Social Chatbots. *arXiv preprint arXiv:1801.01957* (2018).
[59] Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of language and social psychology* 29, 1 (2010), 24–54.
[60] Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-Switched Language Models Using Neural Based Synthetic Data from Parallel Sentences. *arXiv preprint arXiv:1909.08582* (2019).
[61] Donald Winford. 2003. *An introduction to contact linguistics*. Wiley-Blackwell.
[62] Yossi Matias Yaniv Leviathan. 2018. Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone. (2018). https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html
[63] Jennifer Zamora. 2017. I'm Sorry, Dave, I'm Afraid I Can'T Do That: Chatbot Perception and Expectations. In *Proceedings of the 5th International Conference on Human Agent Interaction* (Bielefeld, Germany) *(HAI '17)*. ACM, New York, NY, USA, 253–260. https://doi.org/10.1145/3125739.3125766
[64] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too?. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 2204–2213. https://doi.org/10.18653/v1/P18-1205
[65] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2018. The design and implementation of XiaoIce, an empathetic social chatbot. *arXiv preprint arXiv:1812.08989* (2018).

Received October 2019; revised January 2020; accepted March 2020