Verified Linguistic Style for Conversational Agents

We propose a framework for quantifying linguistic style and explicitly modeling it for conversational agents. It allows agents detect and employ style variations to actively estimate unknown preferences and hidden states of users like formality, personality and language proficiency. This information is used by the agent to customize and adapt to a user, making the experience more engaging for the user. Reasoning about style in interpretable ways enables guarantees and bounds on the tone, manner or style of the agent.

1. Introduction

Even as conversational agents become more and more ubiquitous, they fall short of many aspects of natural conversation that are intuitive for humans. Conversational language has a rich social dimension, and is modulated by the social goals, personalities, affective states and interpersonal relationships of the conversational participants. In human-human conversations, these variables play a critical role in shaping how people express themselves, often manifesting as small differences in the manner or tone or style of the utterance. *"Please pass the salt."* may be perceived as a command or a request depending on whether the word *Please* has any impact in the context, whereas *"Could you please pass the salt?"* is more likely to be perceived as a polite request.

The transmission of meaning in a dialogue depends not only on structural and semantic content of the utterances, but also on the context of the utterance and the inferred intent and state of the speaker. Such inference or implicit feedback in a conversation is useful for the listener to decide on an appropriate response. Humans are adept at getting implicit feedback from the manner of an utterance addressed to them, and using this information to rephrase or modify their response to make it more appropriate. A customer care agent may decide to respond more empathetically if they detect that a customer expressing some frustration. Humans also use conversational style to convey social meaning. Bold language and strong opinions may be used to convey assertiveness, whereas a more agreeable phrasing could be adopted if the speaker intends to come across as polite.

In this paper, we propose a framework for reasoning about such hidden factors or states modulating the expression of language, and using these inferences to make conversational interfaces more natural and engaging for their users. These hidden states could be the emotional state of the user, or their expectation of formality from the agent, or other preferences of the user regarding the bot's behavior. Following the reinforcement learning setup used by (1) in the context of driving, we incentivize our system to actively probe a hidden internal state to improve belief over it. This information is then used to make decisions concerning appropriateness of future utterances, like how to rephrase a candidate response to better suit the tone of the conversation.

In Section 2, we define the distinction between linguistic style and linguistic content. We also enumerate various dimensions of linguistic style, and the hidden variables modulating them. Section 3 formally defines our reinforcement learning framework. It is followed by a description of the procedure of training and using the trained system in Section 4. We then give some concrete applications in Section 5.

2. Linguistic Style

For the purpose of making the framework simpler, we make a distinction between the content of a conversational utterance and its style. Whereas the 'content' of an utterance refers to the literal, semantic aspect of the utterance which is task-relevant, the 'style' of an utterance captures different ways in which the same content can be expressed linguistically, without altering its task-relevant meaning in the context. This variation manifests as a choice among possible syntactic and lexical alternatives, though this choice need not be made consciously by the speaker or the model. We use the word 'style' in the current framework to refer to these utterance-level variations and alternatives, as opposed to discourse-level style (like the topic of the discourse or how the content is organized and presented). A few examples of this variation are as follows * :

- The emotion or sentiment (and its strength) that an utterance is intended to convey.
- How polite or appropriate an utterance is in the context of the conversation.
- If the words used are strong or mild.
- The amount of non-task content or smalltalk introduced into the conversation.
- The complexity and level of difficulty of the vocabulary used.
- Whether an utterance is confident or tentative.
- How analytical the tone of a sentence is.
- What is the information density of the utterance.
- The choice of words among relevant/synonyms based on different senses and affect that they convey.

* This is a tentative list and we will come up with a more exhaustive list after collating any relevant literature

- How opinionated the content of a sentence is.

This linguistic choice both depends upon and conveys some information about various pragmatic factors of the interaction and the preferences and states of the user (2) (3), examples of which are listed below :

- How formal or distant does the speaker intend to be
- What is the language proficiency of the speaker
- What is the interpersonal relationship and social distance between the two parties (4)
- How empathetic the personality of a participant is
- How emotionally expressive a person is

Most conversational agents do not explicitly reason about the factors that dictate the style variations in language expression. They are opaque to this variation. Perceived appropriateness is often sensitive to small stylistic variations, and they fail to make such inferences as above that can inform social appropriateness. A customer facing agent may always have an empathetic or polite tone, and it may be perceived as annoying, but the bot wouldn't be able to change its style based on the feedback. Imparting knowledge to conversational agents about appropriate style decisions can make conversations with them much more engaging for the user, and enhance the overall user experience.

One approach conversational agents take towards style is to completely rely on the input data to dictate/drive the tone and style of their responses (5). This approach is problematic because it doesn't attempt to reason about or control for appropriateness at all, and such systems may go rogue with offensive or undesirable behavior.

Another way current conversational agents regulate style of their utterances is through a blacklisting approach - there is a list of words or phrases that the agent is prohibited from using. Yet another common approach of handling and limiting conversational style is whitelisting - the agent can reply only in one of the many things it is programmed to say, or one of the templates it is allowed to use. Limiting the state space of the dialog to control for unintended consequences also greatly limits expressibility. Another drawback of such systems is that they feel too unnatural to the user, compromising on overall experience and satisfaction and being much less engaging.

All of these approaches are inflexible and coarse. It is our goal to make the language style of the bot more adaptive to the user's own language style and style preferences. Our approach is not only to have an encoding for various style dimensions, but also to make the agent's decisions interpretable. Every style choice of the bot can be traced back to its belief over user preferences, and we wish to provide bounds over what style the bot can and cannot adopt in a scenario, where the bounds are defined over the style parameters the bot learns under supervision.

Our framework assumes that the content and style, as described above, can be modeled separately. We presuppose the existence of an underlying task-oriented, content-driven conversational agent, and build a layer of style reasoning on top of it. This augmentation acts as a style filter over the content generated by the underlying agent, which we henceforth call the content model. The content model decides 'what to say' and the style model figures out 'how to say' it. The style model takes the output of the content model and rephrases it with an appropriate style. However, we do not necessarily assume that the output of the content model would have a neutral style. They may do some implicit style transformations of their own, based on their own training data. The output of the content model may have some style parameters, which will be corrected by the style model based on what it believes is the optimal style according to its own reward function. The workflow is shown in Fig 1, and is explained in more detail in the formalism.

The challenge lies in equipping conversational agents with the ability to incorporate feedback from the user to adapt a style adequate for the setting, and to optimize on task-completion and efficiency without compromising on user experience and perception. This would demand an explicit modeling of human or dialog states during the conversation. By using knowledge of past interactions, this agent should be able to tune its behavior, and also adapt to changing dynamics of the interaction. Another challenge lies in reliably and robustly mapping linguistic utterances in context, to metrics of style measurement, norm adherence, norm violation, etc. It is desirable to also capture the uncertainty in this mapping for ambiguous instances. Pre-defined strategies for social navigation seem too restrictive, so we would like the conversational agent to discover novel and better ones. This is especially useful if the dialog interactions are more than a few turns long, as it allows more room for planning and for long-term strategies to emerge.

Our hope is that by independently reasoning about the style of language and its influence on the users' perception, the framework will help improve user engagement for existing conversational systems. Going forward, the information over user preferences gained by the style model can be used even by the content model in various ways, and thus objectively improve task performance as well.

The framework proposed in this paper can be used to provide guarantees that the agent will abide by certain conversational norms. It can also define and limit the extent and nature of the stylistic variations the agent uses, as an adaptive model parameter that it learns online.

The approach in (Sadigh et al, 2016) (1) is to model unobserved, hidden parameters of human drivers like aggression and attentiveness, and estimate them through inverse reinforcement learning over observed physical movements. By rewarding the robot to actively gain information over the unobserved states, the model learns to perform dis-

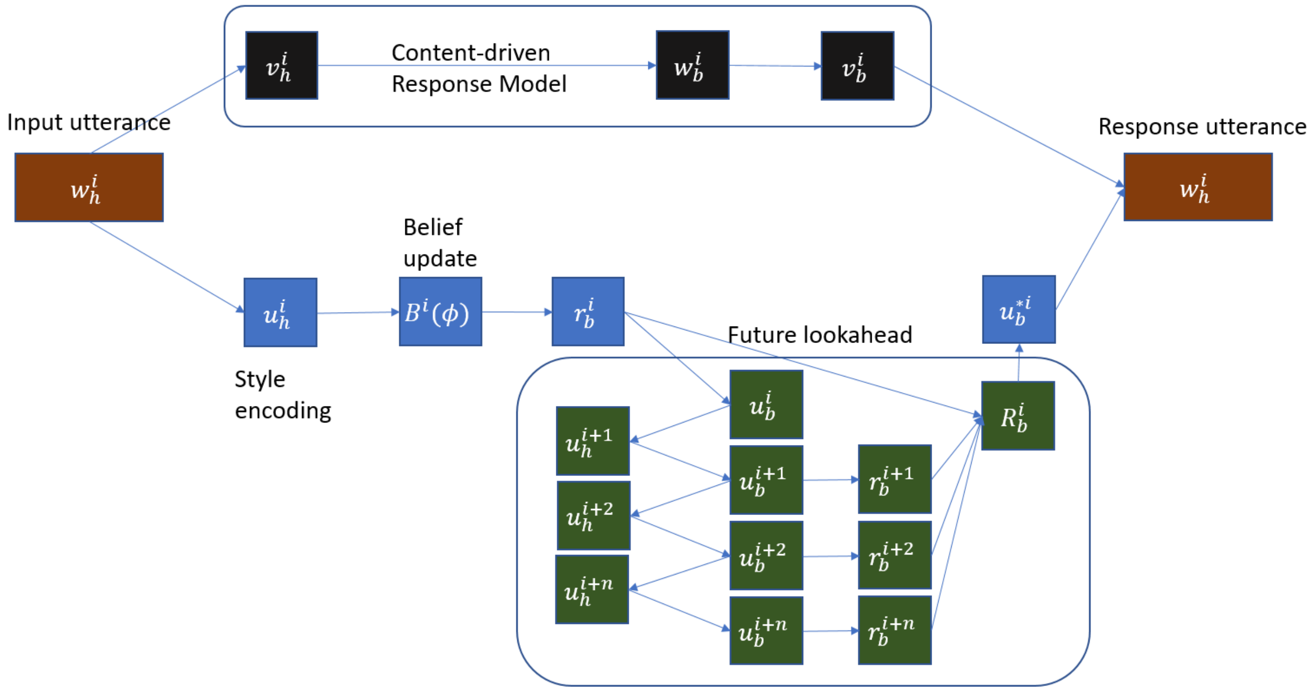

Fig. 1. Model architecture

criminative actions over these states, like nudging into the human's lane to observe the reaction. Analogously, our conversational agent would gain information on the user's preference for degree of formality from their responses, and use this to plan the most appropriate strategies and long-term policy.

3. Formalism

We build upon a formalism which has been proposed for modeling human states in semi-autonomous vehicles (1). The human and the bot take turns producing utterances w_h and w_b, with a turn i consisting of a bot utterance w_b^i followed by a human utterance w_h^i. An utterance w^i is a sequence of tokens produced at turn i by the human or bot, in response to previous utterances.

Let $u \in \Psi$ denote a style encoding of w. The human's conversational style, u_h, is influenced by a hidden internal parameter ϕ. Given $U^{i-1} = (u_b^1, u_h^1, u_b^2, ..., u_h^{i-1})$, the bot outputs u_b^i.

The bot cannot directly observe ϕ, but it has a belief distribution $B^i(\phi)$ that gets revised at every turn.

$$B^i(\phi) \propto B^{i-1}(\phi) \cdot P(u_h^{i-1}|U^{i-2}, u_b^{i-1}, \phi) \qquad [1]$$

The bot's conversational style, u_b, is decided based on a reward function r_b, which rewards, among other things, a reduction in uncertainty in belief $B^i(\phi)$. The reduction in uncertainty is measured by incremental information gain in B :

$$r_b(U^{i-1}, u_b^i) = H(B^{i-1}) - H(B^i) + \lambda \cdot r_{goal}(U^{i-1}, u_b^i, B^i) \qquad [2]$$

The bot thus plans to take discriminating actions - those which are likely to evoke different responses for different values of ϕ. We call this term the exploration reward, as the bot explores the space of future actions. The relative weight of the exploration reward can be changed over time (see A for more details).

The reward has another term that exploits this knowledge over ϕ, such as the bot's accommodativeness to the user's style. We will call the latter reward r_{goal}. Linguistic accommodation or style-matching is a known phenomenon in human-human interactions (6) (7) (8), and the bot can attempt to mimic the user's style in order to make the interaction more natural for the user. If the exploitation reward is the bot's accommodativeness to the user, one of the ways to measure it can be the expected cosine similarity of the response style u_b^i with the predicted human utterance u_h^{*i}, conditioned on B^i:

$$r_{goal}(U^{i-1}, u_b^i, B^i) = \frac{u_b^i \cdot u_h^{*i}}{\|u_b^i\| \|u_h^{*i}\|} \qquad [3]$$

The human is assumed to be maximizing their own reward function r_h^ϕ parametrized by the internal state ϕ. This reward is found using inverse reinforcement learning (9) from demonstrations of human conversations associated with known measurements or annotations of ϕ.

At step i, when computing the optimal encoding u_b^i, the bot maximizes expected reward R_b^i over a finite future horizon. Let $\widetilde{\mathbf{u}}_b = (\widetilde{u}_b^i, ..., \widetilde{u}_b^{i+N-1})$ be a finite sequence of the encodings of the bot's future utterances, and $\widetilde{\mathbf{u}}_h = (\widetilde{u}_h^i, ..., \widetilde{u}_h^{i+N-1})$ be the corresponding human encodings. The reward R_b^i (and the estimated human reward R_h^i) is then a function of the previous encoding u^{i-1}, and the projected future encodings $\widetilde{\mathbf{u}}_b$ and $\widetilde{\mathbf{u}}_h$. The optimal encoding sequence $\widetilde{\mathbf{u}}_b^*$ is given by

$$\widetilde{\mathbf{u}}_b^* = arg \max_{\widetilde{\mathbf{u}}_b} \mathbb{E}_\phi \big[R_b^i(u^{i-1}, \widetilde{\mathbf{u}}_b, \widetilde{\mathbf{u}}_h^{*\phi}(u^{i-1}, \widetilde{\mathbf{u}}_b))\big] \quad [4]$$

where the expectation is taken over the current belief over ϕ, $B^i(\phi)$. $\widetilde{\mathbf{u}}_h^{*\phi}$ are the projected encodings of the human future utterances when the bot executes utterances corresponding to $\widetilde{\mathbf{u}}_b$. These encodings can be approximated under the assumption that the human has a priori access to the bot's projected encodings $\widetilde{\mathbf{u}}_b$:

$$\widetilde{\mathbf{u}}_h^{*\phi}(u^{i-1}, \widetilde{\mathbf{u}}_b) = arg \max_{\widetilde{\mathbf{u}}_h} R_h^{i\phi}(u^{i-1}, \widetilde{\mathbf{u}}_b, \widetilde{\mathbf{u}}_h) \quad [5]$$

Optimizing expected reward entails reasoning about the effects that the bot utterances will have on what observations the bot will get (the style encodings of the utterances that the human produces in response) and how useful these observations will be in shattering ambiguity about phi.

The horizon reward is a simple sum over single-step rewards, for both human and bot :

$$R^i = \sum_{t=i}^{i+N-1} r^t \quad [6]$$

A. Discussion. There are some notable differences from the formalism of (Sadigh et al, 2016) (1). In the absence of a parallel to continuous controls for acceleration and steering, the conversational framework does not have a dynamic control model. Rewards are functions only of the current and previous values of style encodings u of the utterances spoken so far, and of the current belief over the internal state. The future horizon now consists of multiple turns of conversation between the human and the bot, as opposed to a finite window over a single interaction between two cars. Therefore, the size of the horizon and the relative discounting of future rewards may need to be reformulated for the new setting. Another difference from (Sadigh et al, 2016) (1) is the absence of a variable x for the world state. Its equivalent would be a dyad-level style state, which we have separated into the style encodings for the bot u_b and the human u_h.

It is worthwhile to note that the model assumes the internal state ϕ to remain constant throughout the conversation. This assumption may not hold for longer or repeated interactions, depending on what real-world parameter ϕ corresponds to. For instance, the expectation of formality might decrease once some rapport has been established between the two participants. One of the ways in which this limitation of fixed ϕ can be overcome is by attaching higher relative importance to recent utterances, and less to older ones, so that it is easier to shift a belief over ϕ when there is a drift in the underlying value of ϕ. We could also try other ways to enable the model to deal with changing values of ϕ, like by forgetting the belief learned over older utterances.

The goal of the agent is not just to gain information about the user's states, but to use this information to improve its task performance. In other words, r_{goal} consists of any reward that capitalizes on the knowledge of ϕ, and can be closely tied to the application or even the task-specific content model.

We also do not want the agent to compromise too much on other desirable behavior in the interest of gaining information about the user. A competing, 'safety' reward prevents the bot from adopting an incoherent or erratic style for the sole purpose of probing the user. For example, it is particularly important in the first few turns of the interaction for the bot to follow some accepted sociocultural norms. This can be taken care of by adding yet another term to the reward function, whose weight diminishes over time. This reward would be attached to adhering to a formal register.

The relative weights of these rewards would need to be chosen experimentally or through an optimization technique (10).

4. Procedure

In this section, we describe the entire pipeline of the framework, and use a running example of linguistic strategies of politeness described in A to illustrate the broader framework. The training process is described in B, and the online workflow of responding to a user is described in C.

A. Formulation. The style encoding u consists of the various expressions that convey politeness (or rudeness) in language. Examples of such expressions are - apologizing before a request ("*Sorry to bother you ...*"), deference ("*Nice work!*"), and using a positive lexicon ("*That's wonderful!*"). A more comprehensive list of these strategies can be found in (11).

The corresponding internal variable ϕ would capture if the context of the interaction is formal and civil (in which a high degree of politeness is expected from the system) or informal and casual (in which being polite may be perceived as cold or unengaging, and the bot needs to be more casual instead). Without having this information a priori, the bot would plan to discern this expectation of formality from a few interactions with the human.

Since the agent focuses on linguistic style and delegates semantics to a content model, its task now is to come up with a linguistic style plan or strategy to figure out if the interaction is expected to be formal or informal, by taking actions that would evoke different style of responses from the human in the two settings. Future actions of the agent can then use this information to tailor the dialog to suit the specific interaction context.

B. Offline Training. Before any real-world interaction can begin, the agent needs to learn to detect style in utterances, generate utterances with the desired style, and have a human reward model.

B.1. Linguistic Style Understanding. Given a human utterance w_h, produce its style encoding u_h. Depending on what conversational style we are interested in, this mapping may be a rule-based encoding, the output of a style model trained on data, or it could be based on crowd-sourced human judgements. We would also want to normalize style scores based on how salient/marked or unmarked a style is, given the discourse and the demographic context.

In the case of politeness encodings from utterances, we rely on a metric based on crowdsourced politeness judgements (11), where various linguistic strategies are ranked on their perceived politeness ratings.

B.2. Style-Aware Response Generation. Decode a style u_b and a given content encoding v_b in order to generate a valid, coherent utterance in response to the human user. This is quite a challenging task, and for our scenario of linguistic politeness, we follow a simple template-based modification of the base sentence (decoded only from the content model's output v_b) to most closely match the desired u_b. An example of a template would be to prefix a direct request with *"Please"*, add and emphasize words of gratitude, replacing indicative modals like *"will"* and *"can"* with counterfactual modals like *"would"* or *"could"*, etc.

B.3. Human Reward Model. To be able to anticipate human actions and to compute its own reward function, the agent needs to observe multiple real conversations, for which the variable ϕ has been measured, controlled for, or is otherwise known. For more robust training, it is preferable that such observations are made in varied interaction settings, various dialog flows, etc. Using existing human-human conversations for training the style dynamics may not be optimal, since there may not be enough style variations in the utterances to cover the space of style variations. And the style dynamics of human-human conversations would be quite different from those of human-bot conversations.

We propose to collect these observations by making the agent talk at length to multiple people, under settings in which the hidden state ϕ is known. The interactions can be pre-scripted in terms of content and style for the bot, and the bot learns to associate the dynamics of human style u_h corresponding to measurements of ϕ. One simplifying assumption that can be made here is that it is sufficient to collect observations of style choices for humans whose ϕ values lie on extreme ends, i.e. an ideal formal user on one end and an ideal informal user on the other. All intermediate values of ϕ can be considered to be an interpolation of the extreme values. Direct measurements corresponding to the intermediate values of ϕ will add perplexity and may require much more observations of conversation flows.

These observations are then used to recover the human reward function r_h^ϕ using inverse reinforcement learning. This reward would be a weighted linear combination of rewards for individual linguistic strategies, which are also features defined over the space of style encodings:

$$r_h(U^{i-1}, u_b^i, \phi) = \theta(\phi)^T f(U^{i-1}, u_b^i) \quad [7]$$

The weights are deduced under the assumption that the human is optimizing a fixed reward function of the same form, i.e., different human states and interactional contexts differ only in the weights associated with individual features. Since the human reward model is learned from exploratory scenarios of agent-human and human-human interaction, it is inherently aware of the influence that a dialog participant's style has on the other participant.

C. Response Pipeline. Once equipped with a knowledge of human reward functions across different internal parameters ϕ, the agent can now converse with a new human user whose ϕ is unknown. The task of responding to a human utterance w_h involves the following sequence of steps :

1. Map the input human utterance w_h to a style encoding u_h.
2. Update the belief estimate of the hidden variable $B(\phi)$ given the new observation u_h according to eq 1.
3. Compute an expectation of human reward across a future horizon according to eq 5. Note that this expectation is over the current belief of ϕ.
4. Plan actions $\tilde{\mathbf{u}}_b^*$ over the horizon to maximize information gain over ϕ, according to eq 4.
5. Decode the optimal action alongwith the output of the content model v_b to generate the response utterance w_b, according to template-based decoding rules.

Because the bot optimizes rewards over a horizon, the expected behavior is that it plans actions to uncover user preferences, and comes up with novel conversational strategies to do so. After a few turns, the agent should have a higher confidence over the hidden variable ϕ than baseline systems that either only passively estimate ϕ or do not model ϕ at all. Under the assumption that better confidence over ϕ improves the task goal of the underlying system, we could evaluate our framework on the task goal r_{goal}. It would also be worthwhile to evaluate the framework on metrics of user engagement, user satisfaction and ratings of the bot's perceived naturalness. These evaluation could be against the following baselines:

1. A style model that passively estimates ϕ, i.e., only optimizes for r_{goal} without explicitly planning to reduce uncertainty in ϕ.
2. A style model that is not parametrized on ϕ, and is sensitive to language-level style parameters u only. This system has a single reward model for the human.
3. A style model that associates a random style encoding u with each of the responses of the content model. This system has no model for human rewards.
4. No style model on top of the content model.

One of the more sophisticated ways of generating style-aware responses can be a neural generative approach (12). If the underlying content model is a sequence-to-sequence encoder-decoder architecture, then style modifications can be plugged into it as style embeddings analogous to the persona embeddings in (12).

Training sentence-level style embeddings would need supervision on what ways of phrasing a sentence are stylistically similar, and can be looked into in much more detail in a future work. Using style embeddings would however require a departure from interpretable style dimensions in our current formulation of linguistic style encodings. The advantage of using style embeddings is that the model will not be restricted to only those strategies that are fed into it, like the politeness strategies of (11). It would be able to pick up novel features of linguistic style under the right supervision. An obvious disadvantage would be that the dimensions of these embeddings will no longer be directly interpretable, though interpretable paths or contours or manifolds withing the embedding space can be deduced with some effort. The use of style-capturing embeddings is nevertheless an interesting future extension, and would make the model more widely applicable.

5. Applications

The ability to estimate and detect the style of a message can be incredibly useful in many contexts. We pick three use scenarios and describe them in detail - customer satisfaction for troubleshooting assistants, language proficiency and preference for multilingual non-task-driven/chitchat conversational agents, and tone-aware reply suggestions for emails.

A. Multilingual chatbots. Bilingual or multilingual speakers often mix multiple languages, a phenomenon known as code-switching. Code-switching comes in different shapes and sizes, with variations in the extent and complexity with which the languages mix and interleave. A user's preferred extent and nature of code-switching depends not just on their proficiency in each language, but also on socio-linguistic factors like the perceived level of formality of the interaction, and the identity of the participants. Code-switching is a robust phenomenon observed across various multilingual communities. It is an important sociolinguistic phenomenon and a systematic differentiator of an ingroup versus an outgroup (refer to (13) Ch.11).

Multilingual user interfaces, like Microsoft's English-Hindi chatbot Ruuh, and the Indian version of the digital assistant Cortana, can learn to model code-switching as a style variable. To sound natural and agreeable to a user, a multilingual chatbot would want to eventually gravitate towards a code-switching style with which the user is most comfortable. The bot should start the interaction making minimum assumptions about switching (e.g. for an En-Hi bot, start the conversation entirely in English or in Hindi), slowly introduce switching into the conversation, and based on whether the user reciprocates, decide to increase or decrease the extent of switching.

It is important to note that this interaction takes place over multiple conversation turns, and it is not necessary that feedback will be immediate. Moreover, the value of the hidden variable (in this case the desired level of En-Hi switching) can itself change in the course of the conversation. As an example, for an interaction between two En-Hi bilinguals, more code-switching can be expected after a rapport has been established, whereas the initial few interactions would tend to be more formal, and therefore primarily in English. A lot of code-switching in the initial few turns may feel overly friendly and therefore not very appropriate, whereas it may be more appropriate after some rapport has been established.

A.1. Style encoding. Ranging from low to high code-switching, the possibilities are listed below, along with examples from English-Hindi bilingual utterances :

1. No code-switching.

   "Dhanyawaad mujhe paathshaala se ghar tak laane ke liye, warna der ho jaati" (Thank you for dropping me home from school, otherwise it would have been late.)

2. Present only at the level of borrowed words.

   "Dhanyawaad mujhe school se ghar tak laane ke liye, warna late ho jaata"

3. Switching at the level of tags and frozen expressions.

   "Thank you mujhe school se ghar tak laane ke liye, otherwise late ho jaata"

4. Switching at the level of clauses.

   "Thank you for dropping me home from school, warna der ho jaati"

5. Full code-switching.

"Thank you mujhe school se ghar tak drop karne ke liye, otherwise der ho jaati"

The extent of code-switching can be quantified as a vector u_{cs} with a dimension associated with each of the five possibilities listed above. A 0 for a dimension denotes an absence of any switching of that kind, while a 1 denotes that the extent of switching is the highest possible in that dimension given the sentence. There exist metrics for computing the complexity of code-switching like CMI (or code-mixing index) (14) which can be repurposed here as the style encoding.

A.2. Hidden variable. The hidden variable ϕ_{cs} should capture the user's implicit preference for the extent and type of code-switching, which is expressed at the level of the discourse, and estimated from individual utterances. It may be sufficient to model it as a single dimension, with 0 denoting a strong preference for no mixing, and 1 denoting a strong preference for a high level of mixing. The bot would start the interaction with a prior belief of ϕ_{cs} concentrated around 0, and update this belief throughout the conversation towards higher values until an optimum is reached.

A.3. Linguistic Style Understanding. For u_{cs}, the switching style can be measured by a metric based on CMI (14). It is preferable that this score accounts for the base rate of switching provided the content and context. For example, in naturally occurring code-switched En-Hi language, the English word *'school'* is used much more commonly than its Hindi translation *'paathshaala'*, whereas the English word *'house'* is less common than the Hindi *'ghar'*. Therefore, *'paathshaala'* has a higher surprisal than *'ghar'*, and the former should influence the dimensions of u_{cs} much more than the latter. In other words, the word *'paathshaala'* has a much higher markedness of style than the word *'ghar'*, owing to the difference in relative frequency with respect to their respective translations.

A.4. Style-Aware Response Generation. Given the content embedding v_b, we first generate a monolingual sentence w_{mono} from it. We then arrive at a set W_{cs} of all possible code-switched sentences that can be generated from w_{mono} following known linguistic constraints like the Functional Head constraint (15) (16) . Each of the utterances in W_{cs} is scored on the style dimensions of u_{cs}, and the one closest to the optimal is selected, where the optimal style is calculated according to eq 5 over the current belief of ϕ_{cs}.

B. Troubleshooting assistants. Information technology services companies are increasingly adopting automated assistants for customer queries. Human customer care agents are skilled at estimating the level of satisfaction of a customer. A customer may or may not express frustration over their particular grievance or over the agent's responses. They may or may not have a sense of urgency in their request. Automated assistants would greatly benefit from such estimation [†], as it offers the ability to decide a course of action based on the customer's state. The agent would want to decide between an empathetic, reassuring response or a matter-of-fact, task-driven response, or make other task-specific decisions, like deciding if the customer can follow troubleshooting instructions, or if the issue needs escalation.

This is an ideal setting for our framework for many reasons. The interactions are task-driven and yet have a high occurrence and importance of emotional expression. The ideal response is sensitive to the customer's experience, which cannot directly be observed or measured during the interaction.

The style encodings and hidden variables can respectively be linguistic politeness strategies and the formality expectation, as described in Section 4B. It may also be possible to use the style guidelines commonly used to train human troubleshooting assistants, and formulate style dimensions based on it.

C. Email reply suggestions. Another domain of application is email reply suggestions. Based on how an email is phrased, it may be crisp and concise or it may be replete with pleasantries. It may be straightforward or passive-aggressive. The tone may be cold and authoritative or warm and cordial. Current autoresponders like Google Smart Reply for Inbox (17) are good at generating short and diverse responses, but they often do not care about if a reply will be perceived as rude. A model like ours would provide the right sugarcoating to suggest replies with the correct tone. As an example, responding to a question *"Would you be able to attend?"* negatively can be changed from *"No, I won't be."* to *"Sorry, I really wouldn't be able to. But thanks for asking."*. A closely related use-case to smart replies is an assistant that helps you improve your emails before sending them. The assistant informs the user of the style parameters of their text, and suggests ways of rephrasing it according to the desired style.

1. Sadigh D, Sastry SS, Seshia SA, Dragan A (2016) Information gathering actions over human internal state in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on.* (IEEE), pp. 66–73.
2. Gou L, Zhou MX, Yang H (2014) Knowme and shareme: understanding automatically discovered personality traits from social media and user sharing preferences in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems.* (ACM), pp. 955–964.
3. Yarkoni T (2010) Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers. *Journal of research in personality* 44(3):363–373.
4. Danescu-Niculescu-Mizil C, Lee L, Pang B, Kleinberg J (2012) Echoes of power: Language effects and power differences in social interaction in *Proceedings of the 21st international conference on World Wide Web.* (ACM), pp. 699–708.
5. Li J, et al. (2016) Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541.*
6. Giles H, Coupland N, Coupland I (1991) 1. accommodation theory: Communication, context, and. *Contexts of accommodation: Developments in applied sociolinguistics* 1.
7. Danescu-Niculescu-Mizil C (2012) Ph.D. thesis (Cornell University).
8. Dragojevic M, Gasiorek J, Giles H (2015) Communication accommodation theory. *The international encyclopedia of interpersonal communication.*
9. Ziebart BD, Maas AL, Bagnell JA, Dey AK (2008) Maximum entropy inverse reinforcement learning. in *AAAI.* (Chicago, IL, USA), Vol. 8, pp. 1433–1438.

[†] http://www.mattersight.com/wp-content/uploads/2016/09/Heres-What-Personality-Sounds-Like.pdf

10. Vanchinathan HP, Nikolic I, De Bona F, Krause A (2014) Explore-exploit in top-n recommender systems via gaussian processes in *Proceedings of the 8th ACM Conference on Recommender systems*. (ACM), pp. 225–232.
11. Danescu-Niculescu-Mizil C, Sudhof M, Jurafsky D, Leskovec J, Potts C (2013) A computational approach to politeness with application to social factors. *arXiv preprint arXiv:1306.6078*.
12. Li J, et al. (2016) A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.
13. Auer P (2013) *Code-switching in conversation: Language, interaction and identity.* (Routledge).
14. Gambäck B, Das A (2014) On measuring the complexity of code-mixing in *Proceedings of the 11th International Conference on Natural Language Processing, Goa, India*. pp. 1–7.
15. Bhat G, Choudhury M, Bali K (2016) Grammatical constraints on intra-sentential code-switching: From theories to working models. *arXiv preprint arXiv:1612.04538*.
16. Li Y, Fung P (2014) Code switch language modeling with functional head constraint in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. (IEEE), pp. 4913–4917.
17. Kannan A, et al. (2016) Smart reply: Automated response suggestion for email. *arXiv preprint arXiv:1606.04870*.