Enhancing AI Assisted Writing with One-Shot Implicit Negative Feedback

Benjamin Towle[1], **Ke Zhou**[1,2]
[1]University of Nottingham
[2]Nokia Bell Labs
{benjamin.towle, ke.zhou}@nottingham.ac.uk

Abstract

AI-mediated communication enables users to communicate more quickly and efficiently. Various systems have been proposed such as smart reply and AI-assisted writing. Yet, the heterogeneity of the forms of inputs and architectures often renders it challenging to combine insights from user behaviour in one system to improve performance in another. In this work, we consider the case where the user does not select any of the suggested replies from a smart reply system, and how this can be used as *one-shot implicit negative feedback* to enhance the accuracy of an AI writing model. We introduce NIFTY, an approach that uses classifier guidance to controllably integrate implicit user feedback into the text generation process. Empirically, we find up to 34% improvement in ROUGE-L, 89% improvement in generating the correct intent, and an 86% win-rate according to human evaluators compared to a vanilla AI writing system on the MultiWOZ and Schema-Guided Dialog datasets. The code is available at https://github.com/BenjaminTowle/NIFTY.

1 Introduction

The average worker reportedly spends around 23% of their time on reading and answering emails (Mark et al., 2012). To alleviate this burden, there is a growing demand for AI-mediated communication systems to draft and potentially fully automate replies for users. These facilitate faster communication by providing suggestions at different stages of the conversational pipeline. Various modes of interaction exist, each with differing trade-offs: smart reply systems – such as in Gmail (Henderson et al., 2017) or Outlook (Deb et al., 2019) – offer a low-latency solution to dealing with simple requests, using a retrieval-based model to present canned suggested replies to the user which can be clicked instead of requiring manual typing. AI writing models – such as used by Jasper or Grammarly – employ generative architectures to produce more complex replies, but may require additional manual editing and / or prompting from a user to obtain the desired result. The smart reply system may also provide an initial skeleton reply, that an AI writing model can later improve upon (Chen et al., 2019).

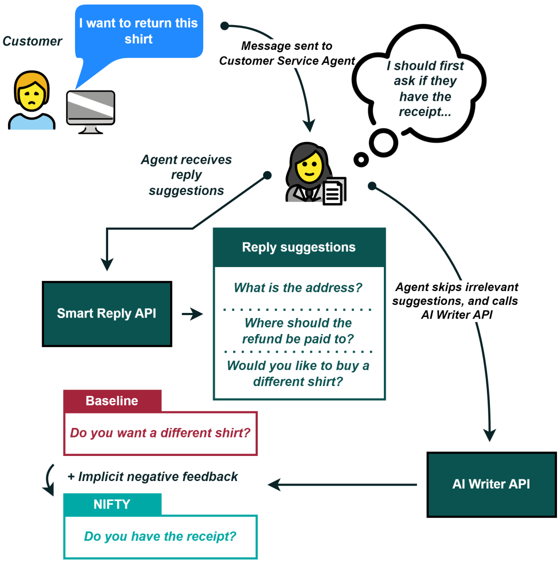
Customer
I want to return this shirt
Message sent to Customer Service Agent
I should first ask if they have the receipt...
Agent receives reply suggestions
Smart Reply API
Reply suggestions
What is the address?
Where should the refund be paid to?
Would you like to buy a different shirt?
Agent skips irrelevant suggestions, and calls AI Writer API
Baseline
Do you want a different shirt?
+ Implicit negative feedback
AI Writer API
NIFTY
Do you have the receipt?

Figure 1: Example of how an agent may utilise either a smart reply system or an AI writing system to speed up communication with a customer. Our approach, NIFTY, uses implicit negative feedback from the rejected suggestions to improve the AI writer's prediction.

Yet, the heterogeneity of the forms of interaction and architectures often renders it non-obvious how the information from one system can be leveraged to improve another: e.g., a smart reply system is only useful when the user clicks one of the suggestions. In practice however, there is often too much uncertainty surrounding the user's intent for any of the suggestions to be relevant (Figure 1) (Chakravarthi and Pasternack, 2017). Due to reading the suggestions, this increases the user's cognitive load, with no additional payoff. This contrasts with positive feedback, when selecting a suggestion reduces user typing time.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 12672–12680
November 12-16, 2024 ©2024 Association for Computational Linguistics

To address this problem, we conduct a pilot study into how *one-shot implicit negative feedback* can be used to integrate and improve the performance between two heterogeneous AI-mediated communication systems. In particular, we consider how feedback when a user clicks *none* of a smart reply system's suggestions can improve a downstream AI writing model at run-time. We concentrate on the one-shot setting, which has the advantage of enabling a single shared model for all users, as well as being more challenging due to the limited amount of interaction information per user. Future work may extend this to greater degrees of user personalisation, although this is currently out of scope due to lack of data access.

In this paper, we introduce **N**egative **I**mplicit **F**eedback from Smar**t** Repl**y** ("NIFTY"). NIFTY employs classifier guidance (Yang and Klein, 2021) to controllably integrate one-shot implicit negative feedback into the generation process at run-time. In particular, given an unconditional AI writing model, we condition the model on a desired attribute c, via an application of Bayes' rule, using a classifier trained to predict c. We explore two possible settings for c: an intent-based approach that conditions on the most likely next intent not represented in the suggestions and a user action-based approach that conditions directly on the user rejecting the suggestions. Overall our method affords several key advantages: (i) it keeps the smart reply and AI writing systems decoupled, allowing it to be readily integrated into existing systems which can be optimised by separate teams; (ii) it enables additional forms of negative feedback to be introduced in the future, e.g., lingering on a suggestion without clicking it (Zhang et al., 2015), via a linear combination with a separate classifier; (iii) incorporating implicit negative feedback into the AI writing process reduces the cost to the user of being presented with irrelevant suggested replies.

Empirically, by evaluating on two publicly available datasets, we find up to 34% improvement in ROUGE-L and 89% improvement in generating the correct intent compared to a vanilla AI writing system, and an 86% win-rate according to human evaluators. In summary, our key contributions are: (1) We introduce the framework of implicit negative feedback to the smart reply and AI writing tasks; (2) We develop and open-source an approach that uses classifier guidance, considering both intent-based and user action-based attributes for conditioning; (3) We provide both quantitative and qualitative analysis of our model's superior performance using both automated and human evaluation.

2 Related Work

Early AI-mediated communication centred around smart reply systems, which provided canned replies to a user message (Kannan et al., 2016; Henderson et al., 2017). Work then expanded to various forms of heterogeneous interaction including AI writing systems, such as autocompletion (Chen et al., 2019), alternative word suggestions (Wang et al., 2023), or conditioning on: rough drafts (Ito et al., 2019), abbreviated sentences (Adhikary et al., 2021), or user provided intents (Sun et al., 2021). While these methods all require explicit user effort, our approach requires no interactive effort from the user, by leveraging implicit negative feedback.

Various types of implicit feedback have been explored in previous works, such as user lingering time (Zhang et al., 2015), skipping content (Pan et al., 2023; Gong and Zhu, 2022), or conversation length (Irvine et al., 2023). To the best of our knowledge, our work is the first to apply this to the smart reply setting.

Finally, our work builds upon the growing literature on controllable text generation. Here, early work focused on conditioning generation on particular control 'codes' (Keskar et al., 2019). More recently, work focuses on classifier guidance, in which a separate classifier guides token-by-token generation (Krause et al., 2020; Yang and Klein, 2021; Shuster et al., 2021; Arora et al., 2022). Note that our focus is *not* to introduce a novel algorithm for classifier guidance, but rather to demonstrate how it can be combined with implicit negative feedback to solve a problem in AI-mediated communication – namely – the lack of easy integration between different modes of interaction. Therefore, as new techniques emerge from this field they may be used to enhance our own method.

3 Method

Filtering with User Simulator We run our user simulator using the dataset $\mathcal{D} = \{(m, r, \mathbf{s})\}_{i=1}^{I}$, where m is the incoming message, r is the ground-truth reply, and $\mathbf{s} = \{s_1, ...s_K\}$ is the set of reply suggestions obtained from a black-box smart reply system. We are concerned with the scenario in which the user clicks *none* of these suggestions. To represent this concretely, let $\mathbf{z}$ represent the set of all possible intents, $\mathbf{z_s}$ be the set of intents assigned to suggestions in $\mathbf{s}$, and z be the intent of the ground-truth response r – e.g., *'Yes I can! Table for 1?'* corresponds to the `booking-inform` intent. We simulate the user by having the user reject all suggestions when $z \notin \mathbf{z_s}$, i.e. none of the suggestions contain the intent of the ground-truth reply. By filtering according to this criterion, we obtain dataset $\mathcal{D}'$ which is used for downstream evaluation. Note that both our generator and classifier are trained on the full version of our dataset $\mathcal{D}$.

Smart Reply The smart reply system uses a vector retrieval model (Karpukhin et al., 2020), trained to jointly embed messages and replies into a shared latent space. At run-time, it retrieves the top K nearest neighbours as suggested replies. Following convention (Deb et al., 2019), we set $K = 3$ for all of our experiments. We choose this straightforward approach in order to demonstrate that even an out-of-the-box smart reply system can provide useful implicit negative feedback, without requiring any bespoke alterations.[1]

Generator Following previous approaches (Sun et al., 2021; Faltings et al., 2023), the AI writing model is a transformer trained to generate tokens autoregressively. Given a message m, the probability of generating reply r can be factorised as:

$$\mathbf{p}_\Theta(r|m) = \prod_{t=1}^{T} \mathbf{p}_\Theta(r_t|m, r_{<t}) \tag{1}$$

This model is trained only on the (message, reply) pairs from $\mathcal{D}$, without requiring access to the smart reply system, or any implicit user feedback, using negative log likelihood.

Classifier Guidance allows an unconditional text generation model $\mathbf{p}_\Theta(r_t|m, r_{<t})$ to generate tokens conditioned on an attribute c, by performing a classification step over possible next tokens $\mathbf{p}_\Phi(c|m, r_{<t})$. These can be combined through Bayesian decomposition (Yang and Klein, 2021).

$$\hat{\mathbf{p}}(r_t|m, r_{<t}, c) \propto \mathbf{p}_\Theta(r_t|m, r_{<t}) \cdot \mathbf{p}_\Phi(c|m, r_{<t}) \tag{2}$$

Note, the classifier predicts whether the attribute *will* be obtained by the time the response is completed, not whether the attribute is currently present. By operating at the token level, rather than over completed generations (i.e. reranking), the model is able to explore a larger search space.

For our purposes, we consider two possible approaches for what attribute to condition the generator on. First, we condition on the desired intent of the response. In particular, we train a classifier to predict the final intent of the reply, given only the reply prefix $r_{<t}$ and message m. At run-time, we then condition on the most likely intent that is not present in the suggestions:

$$\underset{j=1:J}{\operatorname{argmax}}\, \mathbf{p}_\Phi(z_j|m, r_{<t}) \cdot \mathbb{1}_{\mathbf{z}\setminus\mathbf{z_s}}(z_j) \tag{3}$$

where $\mathbf{z} \setminus \mathbf{z_s}$ is the set of intents not present in the suggestions, and $\mathbb{1}$ is the indicator function that outputs 1 if $z_j \in \mathbf{z} \setminus \mathbf{z_s}$, 0 otherwise. The main limitation of this approach is that it requires access to a labelled dataset of intents. To remove this limitation, we therefore also consider conditioning the classifier directly on the user action. Specifically, we attempt to predict whether or not the user rejected the suggested replies as a binary classification task.

4 Experiment

Datasets We evaluate our method on two task-oriented datasets covering various domains: MultiWOZ v2.2 (Budzianowski et al., 2018) and Schema-Guided Dialog (SGD) (Rastogi et al., 2020). Both have the advantage of being annotated with intents, and also feature the professional style of conversations that many AI writing applications aim to facilitate. We treat replies containing multiple intents as unique intents in their own right as it is unclear that a user would accept a suggestion that only partially contained the desired intents. As we are concerned with the assisted writing setting, rather than creating a consumer-facing task-oriented chatbot, we evaluate using standard AI writing metrics, rather than success-based task-oriented metrics. See Appendix A for dataset statistics.

Metrics We employ both automatic and human evaluation. ROUGE-L, which has been used previously in AI writing tasks (Ito et al., 2019), measures the longest common subsequence within the prediction, capturing surface-level overlap with the ground-truth. However, there are often many ways of expressing the same meaning that lack term-overlap. Therefore, inspired by previous evaluation work on smart reply systems (Weng et al., 2019),

[1] While there are at least two open-source implementations for this (Zhang et al., 2021; Towle and Zhou, 2023), we use the latter due to its native support for adding new datasets.

		MultiWOZ		SGD	
Base	Method	R-L	R@1	R-L	R@1
BB	Baseline	25.0	15.6	16.7	28.2
	Unlikelihood	25.3	15.0	16.2	29.1
	Rules-based	25.2	17.5	16.6	25.1
	Reranker Action	24.9	15.4	16.7	28.7
	Reranker Intent	25.2	17.9	16.8	29.2
	NIFTY Action	25.3	18.1	20.4	43.2
	NIFTY Intent	**27.2***	**26.1***	**21.6***	**51.2***
T5	Baseline	27.7	16.4	18.0	28.1
	Unlikelihood	25.3	11.1	17.3	26.1
	Rules-based	27.8	18.0	20.2	30.7
	Reranker Action	24.8	12.8	19.7	35.9
	Reranker Intent	27.6	17.1	19.6	34.8
	NIFTY Action	24.8	12.7	23.8	49.6
	NIFTY Intent	**29.0***	**28.5***	**24.2***	**53.2***

Table 1: Results on MultiWOZ and SGD test sets using ROUGE-L and R@1 metrics. **Bold** indicates best result. * indicates result is statistically significant with p-value < 0.01 compared to best baseline.

		MultiWOZ		SGD	
Model	α	R-L	R@1	R-L	R@1
	0.5	29.4	27.8	24.5	53.1
T5 NIFTY Intent	1.0	29.0	28.5	24.2	53.2
	2.0	27.9	28.9	24.0	53.5

Table 2: Results on MultiWOZ and SGD test sets under different values of α, using the T5 NIFTY Intent model.

			MultiWOZ	SGD
Base	Winner	Loser	Win rate	Win rate
BB	NIFTY Intent	Baseline	73%*	77%*
T5	NIFTY Intent	Baseline	86%*	78%*

Table 3: Human evaluation using crowdworkers. * indicates result is statistically significant with p-value < 0.01 using binomial test.

we complement this metric with R@1, which measures the proportion of generated responses that contain the correct intent. As the responses are generated on-the-fly, we detect intents using a DistilBERT-based classifier trained to map utterances to the corresponding intent for each dataset (a separate classifier to the one used in NIFTY), which we then compare to the ground-truth intent. We conduct human evaluation using Amazon Mechanical Turk. As the purpose of the system is to reduce uncertainty about the user's intended reply, we apply a pairwise setup, in which the evaluator is asked to *select which candidate reply is most similar to the target reply*. For each model and dataset, we randomly select 100 data points from the test set and assign 3 annotators to each data point. We define a 'win' as when a system is achieves a majority vote for a given data point. Overall, the procedure was carried out by 268 unique annotators, across 400 data points. See Appendix C for further details.

Baselines We compare our approach to: the standard *Baseline* AI writing model – i.e. without any classifier guidance; a *Reranker* approach which reranks the final output beams, without any token-level reranking, by selecting the beam with the highest score for the desired intent / action; an *Unlikelihood* decoding approach that downweights the probability for terms that occur in the rejected intents, encouraging the model to generate one of the non-rejected intents (Welleck et al., 2020); a *Rules-based* approach in which multiple candidate beams are generated, and are then filtered to remove beams containing rejected intents from the reply suggestions.

Model Details We explore two different transformers for generation: BlenderBot-400M (BB) (Roller et al., 2020) and T5-220M (T5) (Raffel et al., 2020). At run-time, we generate responses using beam search ($n = 5$). For efficiency, we use the lightweight DistilBERT-66M for classification (Sanh et al., 2019) and only rescore the top 10 tokens with it, following (Shuster et al., 2021). All models are trained with a batch size of 32, learning rate of 5e-5 with linear decay until convergence using the AdamW optimiser. For the generative models, we train using low-rank adaptors (LoRA) (Hu et al., 2022), with an r value of 8, alpha of 32 and dropout of 0.1, see Appendix B.

Main Results Table 1 presents the results for our overall system. We find our approach consistently increases the model's ability to generate the correct intent, with an improvement of up to 89% compared to the baseline. This further corresponds to an improvement in ROUGE-L of up to 34% compared to the baseline. We find that the Reranker and Rules-based approaches fail to improve much upon the Baseline approach, which is consistent with the findings of previous work (Shuster et al., 2021), as the resulting beams typically do not represent a broad range of intents. The Unlikelihood approach also struggles due to different intents often having significant term overlap. Choice of base model proved important for some methods, although ultimately NIFTY Intent remained the strongest ap-

Message	*Sure, 2 people, checking in on Thursday for 3 nights please.*
Sugg. #1	`[Booking-NoBook] [general-reqmore]` Booking was unsuccessful. Would you like to find another hotel?
Sugg. #2	`[Booking-NoBook]` Booking was unsuccessful.
Sugg. #3	`[general-reqmore]` Will you be needing a reference number?
Baseline	I'm sorry, I can't make that for you. Would you like me to try another time?
NIFTY	Your booking was successful. Your reference number is ####. Can I help you with anything else?
Target	`[Booking-Book] [general-reqmore]` Your booking is successful! Your reference number is #### . Can I help you with anything else?

Table 4: Qualitative example from the MultiWOZ test set. Text is post-processed to remove reference numbers as the model does not have access to booking API. Text in square brackets is annotated intents from dataset.

proach in both cases. In terms of choice of classifier, we find the intent-based classifier performs significantly better, especially for MultiWOZ. This gap is much narrower in SGD however, which we hypothesise is due to the smaller number of intents making it easier for the Action classifier to implicitly learn them. Future work may investigate techniques such as unsupervised intent discovery, which has already been used in smart reply to replace the reliance of NIFTY Intent on labelled data (Kannan et al., 2016).

We note also that the classifier used for intent prediction given the message has an R@1 of 47.4% for MultiWOZ and 83.2% for SGD. This difference is due to MultiWOZ having significantly more possible intents (see Appendix A). Future work may explore techniques to improve accuracy such as increasing the context window to multi-turn, as we expect this to improve the overall task performance.

Varying Degree of Guidance In Table 2, we further evaluate how performance for the strongest version of our method, T5 NIFTY Intent, varies under different levels of α, which determines the weighting of the classifier on the logits, i.e., the rightmost term of Equation 1. We find although the approach is broadly flexible to a range of values, higher ROUGE-L scores are associated with weaker levels of guidance, while stronger levels result in higher R@1. Overall, we find the middle-ground setting of 1.0 offers the best trade-off, as well as not requiring any additional hyperparameter search.

Human Evaluation Table 3 shows our human evaluation results. Across both datasets and models, NIFTY Intent is statistically significantly judged better in the pairwise evaluation, with up to an 86% win rate.

Case Study Table 4 presents a qualitative example of model performance. By utilising the fact that the user simulator rejected the unsuccessful booking intent, NIFTY correctly surmises that a successful booking intent is instead required, in contrast to the baseline model, which refuses to make the booking. This example also supports our intuition in designing the user simulator on the assumption that the user would not select a suggestion that only partially overlapped in intent with the ground truth. In this case, although `[general-reqmore]` is present in both suggestions and target, the lack of the `[booking-book]` intent appears critical to appropriately responding to the message.

5 Conclusion

In this work we introduce NIFTY, an AI writing system that uses classifier guidance to account for implicit negative user feedback from an upstream smart reply system. Empirically, we find up to 34% improvement in relevance and 89% improvement in generating the correct intent compared to a vanilla AI writing system. Future work may explore applying these techniques to real-life data from deployed systems and / or may be extended to a broader range of types of feedback beyond click data from smart reply systems.

Acknowledgements

We thank the reviewers for their helpful feedback and suggestions. This work is partly supported by the EPSRC DTP Studentship program. The opinions expressed in this paper are the authors', and are not necessarily shared/endorsed by their employers and/or sponsors.

Limitations

We identify three main limitations of our work, which we address here. First, our work assumes the main driver of a user clicking a suggestion is whether it matches their intended intent. However, there may be additional factors that influence which suggestions a user clicks, such as formality of the utterance, the user's own preference for using AI generated content, or the user's own writing style. However, deployed systems may circumvent this issue by training directly on user click through data, in place of the user simulator used here. Second, the datasets used are somewhat artificial, being deliberately designed as dialogue benchmarks, rather than organic datasets created in an actual working environment, which are potentially more noisy. Third, the most effective version of our method, NIFTY Intent requires a dataset of intent annotations to train the classifier. We regard removing this limitation, such as via unsupervised intent detection, as a possible avenue for future work.

Ethical Considerations

Various ethical concerns have been raised around the potential for generative dialog systems to produce inappropriate content, particularly as their fluency increases and their content because less distinguishable from a human's. However, there is additional nuance in the context of AI-assisted writing in that humans have oversight over the content being generated, and may reject it if it is inappropriate. On the one hand, this may be seen as a mitigating factor, as the human may function as an implicit safety classifier. On the other hand, recent research indicates that suggestions may influence user behaviour to a degree; specifically, while smart reply systems have long been known to have a positivity bias (Kannan et al., 2016), recent work finds that this can influence the behaviour of the system's users. In particular, Wenker (2023) find that users of smart reply systems produced overly more positive sentiment replies than users without access to these systems. Further research is needed on understanding in what other ways assisted writing systems shift the distribution of user replies.

References

Jiban Adhikary, Jamie Berger, and Keith Vertanen. 2021. Accelerating text communication via abbreviated sentence input. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6574–6588, Online. Association for Computational Linguistics.

Kushal Arora, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. 2022. Director: Generator-classifiers for supervised language modeling. In *AACL*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Nimesh Chakravarthi and Jeff Pasternack. 2017. Building smart replies for member messages. press release. `https://engineering.linkedin.com/blog/2017/10/building-smart-replies-for-member-messages`.

Mia Xu Chen, Benjamin Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M. Dai, Z. Chen, Timothy Sohn, and Yonghui Wu. 2019. Gmail smart compose: Real-time assisted writing. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Budhaditya Deb, Peter Bailey, and Milad Shokouhi. 2019. Diversifying reply suggestions using a matching-conditional variational autoencoder. In *North American Chapter of the Association for Computational Linguistics*.

Felix Faltings, Michel Galley, Kianté Brantley, Baolin Peng, Weixin Cai, Yizhe Zhang, Jianfeng Gao, and Bill Dolan. 2023. Interactive text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4450–4468, Singapore. Association for Computational Linguistics.

Shansan Gong and Ke Zhu. 2022. Positive, negative and neutral: Modeling implicit feedback in session-based news recommendation. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Robert P. Irvine, Douglas Boubert, Vyas Raina, Adian Liusie, Ziyi Zhu, Vineet Mudupalli, Aliaksei Korshuk, Zongyi Joe Liu, Fritz Cremer, Valentin Assassi, Christie-Carol Beauchamp, Xiaoding Lu, Thomas Rialan, and William Beauchamp. 2023. Rewarding chatbots for real-world engagement with millions of users. *ArXiv*, abs/2303.06135.

Takumi Ito, Tatsuki Kuribayashi, Hayato Kobayashi, Ana Brassard, Masato Hagiwara, Jun Suzuki, and Kentaro Inui. 2019. Diamonds in the rough: Generating fluent sentences from early-stage drafts for academic writing assistance. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 40–53, Tokyo, Japan. Association for Computational Linguistics.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Gregory S. Corrado, László Lukács, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart reply: Automated response suggestion for email. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq R. Joty, Richard Socher, and Nazneen Rajani. 2020. Gedi: Generative discriminator guided sequence generation. In *Conference on Empirical Methods in Natural Language Processing*.

Gloria Mark, Stephen Voida, and Armand V Cardello. 2012. "a pace not dictated by electrons": an empirical study of work without email. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Yunzhu Pan, Nian Li, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2023. Learning and optimization of implicit negative feedback for industrial short-video recommender system. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric Michael Smith, Y.-Lan Boureau, and Jason Weston. 2020. Recipes for building an open-domain chatbot. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. 2021. Am i me or you? state-of-the-art dialogue models cannot maintain an identity. In *NAACL-HLT*.

Simeng Sun, Wenlong Zhao, Varun Manjunatha, Rajiv Jain, Vlad Morariu, Franck Dernoncourt, Balaji Vasan Srinivasan, and Mohit Iyyer. 2021. IGA: An intent-guided authoring assistant. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5972–5985, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benjamin Towle and Ke Zhou. 2023. End-to-end autoregressive retrieval via bootstrapping for smart reply systems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7610–7622, Singapore. Association for Computational Linguistics.

Chenshuo Wang, Shaoguang Mao, Tao Ge, Wenshan Wu, Xun Wang, Yan Xia, Jonathan Tien, and Dongyan Zhao. 2023. Smart word suggestions for writing assistance. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11212–11225, Toronto, Canada. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Yue Weng, Huaixiu Zheng, Franziska Bell, and Gökhan Tür. 2019. Occ: A smart reply system for efficient in-app communications. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Kilian Wenker. 2023. Who wrote this? how smart replies impact language and agency in the workplace. *Telematics and Informatics Reports*, 10:100062.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, pages 3511–3535, Online. Association for Computational Linguistics.

Aston Zhang, Amit Goyal, Weize Kong, Hongbo Deng, Anlei Dong, Yi Chang, Carl A. Gunter, and Jiawei Han. 2015. adaqac: Adaptive query auto-completion via implicit negative feedback. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Mozhi Zhang, Wei Wang, Budhaditya Deb, Guoqing Zheng, Milad Shokouhi, and Ahmed Hassan Awadallah. 2021. A dataset and baselines for multilingual reply suggestion. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1207–1220, Online. Association for Computational Linguistics.

A Dataset Details

Table 5 presents the detail for the MultiWOZ and SGD datasets. We filter the datasets to include only those examples where the suggestions were rejected in the user simulator, i.e. none of the suggestions had a shared intent with the ground-truth reply. The datasets contain a large number of intents, as we treat replies containing multiple intents as standalone intents.

B Training Hyperparameters

Table 6 summarises the training hyperparameters. To reduce training time for the generator, we use LoRA (Hu et al., 2022). All classifiers used in the paper use the same hyperparameters. We train the classifier for more epochs than the generator as the classifier requires learning the weights for the individual classes in the classifier head from scratch. All remaining unstated hyperparameters are the default provided by the HuggingFace `TrainingArguments` class.

C Human Evaluation

For human evaluation, workers were provided with the the following short instructions: *You will be shown a target reply and two candidate replies. Select which candidate reply is most similar to the target reply.* They were also provided with the following long instructions: *You will be shown a target reply and two candidate replies. Select which candidate reply is most similar to the target reply. Similar means having the same semantic meaning.*

D Disclosure

GitHub Copilot was used to a limited extent for boilerplate code autocompletion. All models and the dataset used in this paper are freely available for use in research.

	MultiWOZ		SGD	
	Pre-filtering	Post-filtering	Pre-filtering	Post-filtering
Train size	56.8k	36.9k	165.0k	30.2k
Validation size	7.4k	4.9k	24.4k	5.2k
Test size	7.4k	4.9k	42.3k	9.8k
# Intents	685		21	

Table 5: Statistics for the MultiWOZ v2.2 (Budzianowski et al., 2018) and SGD (Rastogi et al., 2020) datasets, indicating number of samples before and after filtering to include only samples with negative feedback (i.e., rejected smart replies). Number of intents is large as multi-intent replies are treated as unique intents.

	MultiWOZ			SGD		
	BB-Generator	T5-Generator	Classifier	BB-Generator	T5-Generator	Classifier
Batch size	32					
Learning rate	5e-5					
Learning rate decay	linear					
Warmup steps	100					
Epochs	1	5	10	1	5	5
LoRA settings						
r	8		–	8		–
alpha	32		–	32		–
dropout	0.1		–	0.1		–

Table 6: Training hyperparameters for the BlenderBot (Roller et al., 2020) and T5 (Raffel et al., 2020) generators and DistilBERT (Sanh et al., 2019) classifiers.