Model-Based Simulation for Optimising Smart Reply

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

Abstract

Smart Reply (SR) systems present a user with a set of replies, of which one can be selected in place of having to type out a response. To perform well at this task, a system should be able to effectively present the user with a diverse set of options, to maximise the chance that at least one of them conveys the user's desired response. This is a significant challenge, due to the lack of datasets containing sets of responses to learn from. Resultantly, previous work has focused largely on post-hoc diversification, rather than explicitly learning to predict sets of responses. Motivated by this problem, we present a novel method SIMSR, that employs model-based simulation to discover high-value response sets, through simulating possible user responses with a learned world model. Unlike previous approaches, this allows our method to directly optimise the end-goal of SR–maximising the relevance of at least one of the predicted replies. Empirically on two public datasets, when compared to SoTA baselines, our method achieves up to 21% and 18% improvement in ROUGE score and Self-ROUGE score respectively.

1 Introduction

Automated response suggestion, or Smart Reply (SR), is rapidly becoming a staple feature of many email and chat systems such as Gmail, Skype, Outlook, Microsoft Teams, LinkedIn and Facebook Messenger. Given a message, SR systems present the user with a selection of possible responses, e.g. `How are you?` → {`I'm good`; `I'm ok`; `Not great`}, which they can click in place of having to type out a reply. With the growth of communication over smaller devices that are poorly suited for manual typing (Varcholik et al., 2012; Palin et al., 2019), such as smartphones and smart watches, SR is becoming an increasingly more important feature.

While early methods in SR incorporated sequence-to-sequence models (Kannan et al., 2016), the current mainstream approach favours *Matching models* which separately encode the message and reply into a shared latent space and retrieve the nearest neighbour response (Deb et al., 2019; Zhang et al., 2021; Deb et al., 2021). This has advantages in a production context, as it enables the model to retrieve replies from a fixed response set, maintaining greater controllability of model outputs; further, the latent representations for the response set can be pre-computed prior to inference, enabling faster latency.

However, the naive approach of simply retrieving top-K highest-scoring candidates from the Matching model often fails to produce a sufficiently diverse set of reply options. For instance, in response to the message `How are you?`, if the first predicted response is `I'm good`, predicting `I'm doing well` as the second response provides limited incremental value, as it carries equivalent semantic meaning. By contrast, `Not great` would be more useful, as it captures an alternative semantic meaning a user might wish to convey. In summary, one must account for the *interdependencies* between replies. Previous methods have sought to implicitly account for these interdependencies such as through clustering by intent/topic, learning latent variables or re-scoring replies to include inter-reply similarity (Kannan et al., 2016; Deb et al., 2019, 2021). However, these techniques face two limitations: (1) they require hard-coded trade-offs between message-reply relevance and inter-reply diversity; (2) jointly optimising these two metrics is only partially correlated with the end goal of SR–maximising the relevance *at least one* of the predictions. Ideally, it would be more principled if the model could simply optimise over this end goal. In so doing, we hypothesise performance would improve, while a good amount of diversity should also naturally emerge, insofar as it is correlated with performance on the task.

However, directly optimising this metric

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 12030–12043

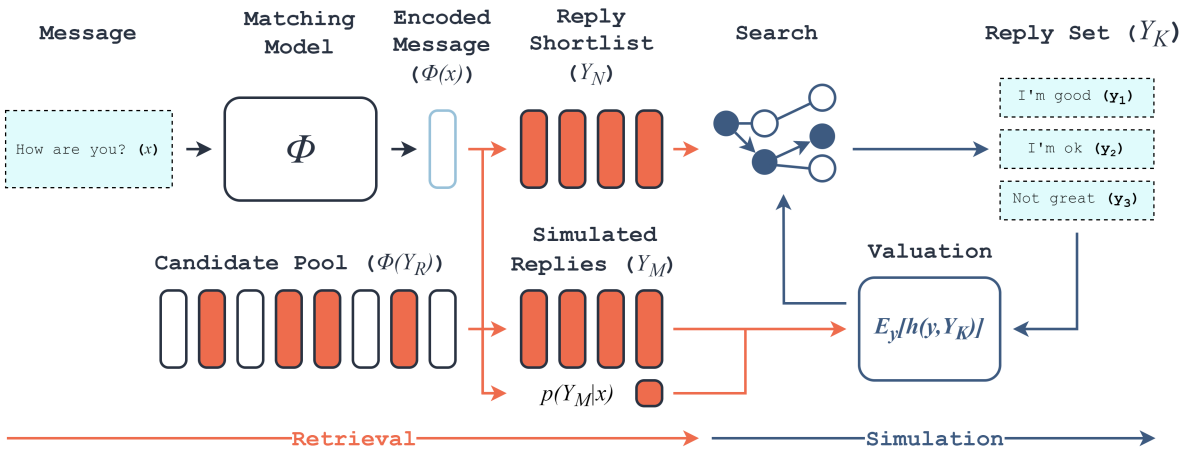

Figure 1: Overview of our approach. We combine a retrieval stage, which obtains the initial reply shortlist Y_N, followed by a simulation stage, which iteratively searches for reply sets Y_K from that shortlist, and evaluates their relevance against a set of simulated replies Y_M.

presents two problems: (1) the probability distribution over replies given messages is initially unknown; (2) we only have access to a *single* reply for each message sampled from this distribution–i.e. the dataset of ⟨message, reply⟩ pairs–which prevents simply learning to predict reply sets via supervised learning. To circumvent these problems, we introduce model-based simulation (MBS) to the SR setting as a possible avenue forward. MBS is a technique from reinforcement learning (Sutton and Barto, 2005) that allows an agent to choose what action to take by simulating the potential consequences of an action using a learned world model. We observe that the Matching model, given it is trained on a dataset of ⟨message, reply⟩ pairs, can also operate as a world model. This allows us to estimate the expected relevance of any reply set, by running repeated simulations with the world model. Crucially, relevance here can be defined as the maximum similarity between the reply set and a response sampled from the world model, which replaces the reliance on hard-coded trade-offs between message-reply relevance and inter-reply similarity.

Concretely, our method–SIMSR (Figure 1)–comprises an initial retrieval stage, followed by an iterative simulation stage. We first retrieve a shortlist of replies from a larger candidate pool, using a learned neural Matching model, conditioned on a given message. In parallel, we also retrieve a number of simulated replies using the same method. Next, for the simulation stage, we use a search module to select a reply set comprising three responses from the shortlist. Then, we use a valuation module, which computes the expected similarity between the simulated replies and the most similar response from the reply set. This can be computed through a simple marginalisation process, using the probabilities and corresponding simulated replies provided by the world model. This process of search and valuation is iterated until the search algorithm terminates, and finally returns the highest scoring reply set. Quantitatively, our experiments show consistent out-performance against existing SoTA methods across two relevant datasets–Reddit and PERSONA-CHAT–achieving up to 21% and 18% improvement in ROUGE score and Self-ROUGE score respectively. SIMSR also runs at a comparable speed to other methods, because the simulation is highly parallelisable and the Matching model only needs to encode the message once for both its initial retrieval and world model roles. In summary, our key contributions are:

- We present model-based simulation as a novel paradigm for the Smart Reply task.
- We present SIMSR, a novel method that employs model-based simulation with a learned world model.
- We demonstrate empirically the importance of taking into account reply interdependencies, achieving SoTA performance across the Reddit and PERSONA-CHAT datasets.

We make our code available for reproducibility.[1]

[1] https://github.com/BenjaminTowle/SimSR

2 Related Work

Smart Reply. In industry, SR has a range of applications from email systems to instant messaging. Naturally, the data from these is not publicly available to train on. Instead, recent work has made use of publicly available dialogue datasets such as Reddit (Deb et al., 2021; Zhang et al., 2021), which is sufficiently similar given SR applications are principally concerned with dialogue. While the earliest SR systems used sequence-to-sequence models (Kannan et al., 2016), nowadays retrieval methods prevail which select a response from a pre-defined pool of candidates (Henderson et al., 2017), i.e. Matching models. By itself however, the Matching model has no way to ensure that the chosen reply set is sufficiently diverse. One approach to this is to ensure that no two responses in the reply set share the same topic/intent (Kannan et al., 2016; Chakravarthi and Pasternack, 2017; Weng et al., 2019). However, this becomes more difficult in an open-domain setting, where the range of topics/intents is difficult to pre-define. As a result, other approaches have focused on more fine-grained diversification through conditional variational autoencoder techniques, which learn topics/intents across a continuous latent space during training (Zhao et al., 2017; Deb et al., 2019). Maximum marginal relevance, which re-weights responses according to how similar they are with one another, has also been shown to work well (Carbonell and Goldstein-Stewart, 1998; Deb et al., 2019). Our method differs from these approaches in that they employ diversity in a post-hoc manner which does not directly optimise the end goal of SR–maximising the relevance of at least one of the predicted replies.

Simulation in NLP. In board games such as Go and chess, a model can have access to a perfect simulator, allowing it to explore various counterfactual trajectories before deciding what action to take next (Silver et al., 2017). In user-facing NLP applications, this is rarely possible. Therefore, much work has focused on settings such as self-play, in which a model learns to become better at a task such as negotiating (Lewis et al., 2017) or even open-domain dialogue (Li et al., 2016a) through interacting with another copy of itself (or a version with frozen weights). User simulators are especially prevalent in task-oriented dialogue, where the domain is narrower and it is therefore easier to anticipate user behaviour (Li et al., 2016b). A notable exception to the above cases is text-based games–scripted games involving interacting in a wholly text-based environment–which are typically trained with access to a perfect simulator, as the game engine allows for previous states to be restored (Jang et al., 2021). Our work is closest in spirit to those works that perform dialogue rollouts to select the next utterance using a reply prediction model (Lewis et al., 2017; Li et al., 2016a)–i.e. the Matching model. However, in our case the rollouts only involve a single step look-ahead, while our action space is the set of possible reply sets, rather than individual utterances. Further, our method can be used out-of-the-box during inference, without any further retraining of the Matching model. So far as we are aware, our work is the first to apply this concept of simulation to the SR setting.

3 Framework

3.1 Task Definition

Our task is to predict a set of K replies $Y_K = \{y_k\}_{k=1}^{K}$ from a candidate pool Y_R of size R, conditioned on a message x. While in an online setting, the aim might be to maximise click-through rate (Deb et al., 2019), in an offline setting this can be approximated as maximising the similarity function $f(y)$, given as the maximum similarity between Y_K and the ground truth response y (Zhang et al., 2021):

$$f(y) = \max_k [\{\text{sim}(y, y_k)\}_{k=1}^{K}] \quad (1)$$

3.2 Matching Model

Following previous approaches, we use a Matching model as the backbone of our method (Henderson et al., 2017; Zhang et al., 2021). This comprises two parallel pre-trained transformer encoders Φ (with shared weights) that *separately* encode x and y into a shared latent space. This is obtained by taking the output hidden-state corresponding to the `[CLS]` token which is pre-pended to each of the inputs. We refer to the vector representations of the message and reply as $\Phi(x)$ and $\Phi(y)$ respectively, and their score $g(x, y) = \Phi(x) \cdot \Phi(y)$. The model is trained using negative log-likelihood to maximise the joint probability of the context and reply:

$$p(x_i, y_i) = \frac{e^{g(x_i, y_i)}}{\sum_{y_j} e^{g(x_i, y_j)} + \sum_{x_j} e^{g(x_j, y_i)} - e^{g(x_i, y_i)}} \quad (2)$$

This is referred to as *symmetric loss* (Deb et al., 2019), and is known to impose tighter constraints

on the relation between the message and reply, compared to having only a one-way classification loss function.

4 SimSR

For any given message x, there is uncertainty about the response y, which we assume to be sampled from some distribution Y. This is commonly referred to as the one-to-many problem (Zhao et al., 2017; Towle and Zhou, 2022) and is due to several reasons, such as unknown facts about the user and their intent. For example, the reply to `Can you meet for coffee at 2pm?` is likely to be conditioned on factors such as the user's schedule or their interest in meeting, which is unknown to a vanilla SR system. As a result, Matching models that simply select the most likely individual replies only achieve a lower bound of potential performance. This can be represented by the following inequality:

$$E_{y\sim Y}[f(Y)] >= f(E_{y\sim Y}[Y]) \quad (3)$$

where $f(Y)$ refers to the similarity function from Equation 1. The right hand side of Equation 3 represents what a Matching model approximates, while the left hand side is what we would like to obtain. Intuitively, this means that a good model should make predictions that capture the range of possible responses that could be sampled from Y, rather than simply the single most likely response. To do this, we hypothesise it is important to develop a method that accounts for the interdependencies between replies, i.e. which can evaluate sets of replies, rather than only individually scoring replies.

Algorithm 1 and Figure 1 overview our method, which can be applied directly during inference. The Matching model first retrieves a shortlist of N replies from a pool of pre-computed candidates Y_R (Section 4.1). Then we combine a search module which selects and constructs reply tuples from this shortlist to evaluate (Section 4.4) and a valuation module (Section 4.3) which computes an expected score between a given reply set and a list of simulated replies (Section 4.2). Note that as our method does not require learning any new parameters, it can be applied to reply sets of arbitrary sizes during inference.

4.1 Reply Shortlist

Given an overall candidate pool of size R, the corollary action space of K-tuples is intractably large: $\frac{R!}{K!(R-K)!}$. To mitigate this, we follow previous work (Deb et al., 2019) and first retrieve the top-N ranking replies conditioned on the message x, using the Matching model, where $N << R$. We refer to this set as $Y_N = \{y_n\}_{n=1}^{N}$. This defines the building blocks with which we can construct the action space of K-tuples of replies to perform our simulation on.

4.2 Simulated Replies

We do not have access to the ground-truth data-generating distribution–i.e. $p_{human}(y|x)$–which would be required for planning in the actual environment. However, the Matching model can serve as an effective approximator of this distribution–henceforth, $p_{model}(y|x)$–since it was trained on ⟨message,reply⟩ pairs sampled from the ground-truth distribution. Thus, using the same Matching model as above, we retrieve the top-M replies, also conditioned on the message x, to obtain $Y_M = \{y_m\}_{m=1}^{M}$. In practice, as we use the same model to retrieve both Y_N and Y_M, this can be achieved with a single query of the response set–therefore, the impact on latency is kept to a minimum.

4.3 Valuation

We define similarity between a K-tuple and the m-th simulated response $y_m \in Y_M$ as:

$$h(y_m, Y_K) = \max_k \{\text{sim}(y_m, y_k)\}_{k=1}^{K} \quad (4)$$

where $\text{sim}(\cdot,\cdot)$ is a similarity score. Intuitively, this rewards the model if at least one of the predictions is relevant to the user. We use term-level F1-score to represent similarity for simplicity, and leave alternative measures for future work. We obtain the expected similarity for a given K-tuple by marginalising over the scores for all $y_m \in Y_M$:

$$E[h(y, Y_k)] = \sum_{m}^{M} h(y_m, Y_K) \cdot p_{model}(y_m|x) \quad (5)$$

In practice, we found dividing the scores by a high temperature ($\tau = 10$) (Hinton et al., 2015) before applying a softmax normalisation improved performance, as it encouraged the model to take into account a larger range of possible simulated responses.

4.4 Search

Given our method for estimating the value of any given K-tuple, it is necessary to employ a search

Algorithm 1 Model-Based Simulation with Ablative Search

Input Matching model Φ, message x, response pool Y_R, number of candidates N, number of simulations M, final reply set size K.
Output reply set Y_K

procedure MODELBASEDSIMULATION(Φ, x, Y_R, N, M, K)
  $Y_N, Y_M, P_M \leftarrow$ RETRIEVE(Φ, x, Y_R, N, M) ▷ Retrieve responses from Y_R with corresponding probabilities P_M.
  $C \leftarrow$ COMPUTESIMILARITY(Y_N, Y_M) ▷ Obtain similarity matrix.
  while $len(Y_N) > K$ **do** ▷ Ablative search loop
    $L, bestScore, bestIdx \leftarrow$ LEN(Y_N), -1.0, $None$
    for $l \leftarrow 0$ to L **do**
      $C_{tmp} \leftarrow$ CONCATENATE($C[:l], C[l+1:]$)
      $eScore \leftarrow$ SUM($P_M \cdot \max(C_{tmp}, axis = 0)$)
      if $eScore > bestScore$ **then** ▷ Assign new best score if needed.
        $bestScore, bestIdx \leftarrow eScore, l$
      end if
    end for
    delete $Y_N[bestIdx], C[bestIdx]$ ▷ Remove least useful reply
  end while
  $Y_K \leftarrow Y_N$
  return Y_K
end procedure

algorithm, to decide which tuples should be evaluated. In this work, we consider a selection of out-of-the-box and bespoke methods:

Exhaustive Search. A straightforward approach is to simply enumerate and evaluate all possible tuples. This is feasible because (a) N is typically a relatively small number (15, in our experiments), (b) the computational cost for evaluating any given tuple is low, given it involves simply computing Equation 5 where the similarity function $\mathrm{sim}(\cdot,\cdot)$ only needs to be computed once for each y_n,y_m pair.

Ablative Search. For larger values of N, it is necessary to employ a more selective search strategy. We observe that the task of finding K replies from a shortlist of N replies can be treated partially as a clustering problem, where each reply in the K-tuple represents a cluster nucleoid, and the objective is to minimise some distance matrix. To this extent, we design a method that incrementally builds the reply set by iteratively removing (hence, *ablative*) the least useful reply from the shortlist N, until only K replies remain. In detail, for each of the $(N-1)$-tuples of Y_N we compute $E[h(y, Y_{N-1})]$, such that Y^*_{N-1} is the $(N-1)$-tuple that obtained the highest score. We then remove the sole reply $y*$ from Y_N that is not present in Y^*_{N-1}. Finally, we repeat this process for all of the $(N-2)$-tuples of Y_{N-1} etc. until we are left with $Y_{N-(N-K)} = Y_K$.

Greedy Search. A limitation of ablative search is that it requires a lot of non-parallelisable compute due to the iterative nature of the algorithm. We therefore consider a greedy alternative. In brief, instead of obtaining Y_K by whittling down Y_N, we instead incrementally build up Y_K starting from the empty set. This thus requires only K non-parallelisable steps, rather than $N-K$. In detail, let Y_G be the set of currently chosen replies, such that initially $Y_G = \varnothing$. Then, for each reply $y_n \in Y_N$ we compute the expected similarity for the union of Y_G and y_n, i.e. $E[h(y, Y_G \cup y_n)]$. Next, we append the highest scoring y_n to Y_G, and repeat until $|Y_G| = K$.

Sample and Rank. Finally, we consider a simple sample and rank approach, which has been shown to work well in other NLP tasks such as dialogue (Freitas et al., 2020). This involves randomly selecting a subset of all possible tuples, and evaluating them. Then, we return the tuple with the highest score according to Equation 5.

5 Experiments

We now turn our attention towards empirical testing of SIMSR, addressing the following research questions:

- **RQ1:** How does the choice of search strategy impact relevance and diversity in SIMSR? (Section 5.5)
- **RQ2:** How does SIMSR compare to existing SoTA SR methods? (Section 5.6, 5.8)
- **RQ3:** How much does SIMSR benefit from accounting for interdependencies between replies when selecting a reply set? (Section 5.7)

5.1 Baselines

We identify four types of diversification strategies which serve as baselines against our model. The

Search	Reddit		PERSONA-CHAT		# Tuples
	ROUGE ↑	Self-ROUGE ↓	ROUGE ↑	Self-ROUGE ↓	Evaluated ↓
Exhaustive	2.47	2.49	**7.85**	8.60	455
Ablative	2.40	**2.36**	7.71	**8.39**	114
Greedy	**2.49**	2.77	7.82	9.76	42
Sample-and-Rank	2.39	2.79	7.39	12.27	**25**

Table 1: Results on the Reddit and PERSONA-CHAT Test sets under different search strategies for SIMSR.

original implementations of these methods are typically proprietary and unavailable for direct comparison. Therefore, in the list below we summarise our re-implementations as well as key changes that were made versus the original.

Matching is the base retrieval model discussed earlier (Section 3.2) (Henderson et al., 2017; Zhang et al., 2021). It simply selects the top-K responses according to their individual scores without any additional components. Our version uses the DistilBERT model as a base (Sanh et al., 2019), whereas previous methods used a variety of transformers (Zhang et al., 2021) and recurrent neural networks (Deb et al., 2019)–we follow this for all baselines.

Matching-Topic uses topic classification to ensure none of the top-K responses share the same topic (Kannan et al., 2016; Chakravarthi and Pasternack, 2017; Weng et al., 2019). We replace the classifier with an out-of-the-box classifier trained on Twitter (Antypas et al., 2022), which features similarly short-form messages to those used in SR.

Maximum Marginal Relevance (MMR) reweights responses according to how similar they are with one another, which is combined in a linear combination with their message-response score (Deb et al., 2019). Our re-implementation is closer to the original algorithm (Carbonell and Goldstein-Stewart, 1998) in that we incrementally build the reply set, rather than in a single step–we found this performed better during early testing.

MCVAE (Deb et al., 2019) is a conditional variational autoencoder (Zhao et al., 2017) built on top of the Matching model, allowing for multiple query vectors to be generated from a single message embedding. Candidates are scored using a voting process whereby each query vector selects the nearest reply, and the K most-selected replies are chosen. We re-implement this without any major changes from the original to the best of our knowledge, and use the original paper's hyperparameters, such as size of the latent variable, where possible.

	Reddit			PERSONA-CHAT		
	Train	Valid	Test	Train	Valid	Test
# Samples	50k	5k	5k	66k	8k	8k

Table 2: Statistics for the datasets.

5.2 Datasets

We evaluate our methods across two datasets, summarised in Table 2. While most prior work has used proprietary datasets (Kannan et al., 2016; Deb et al., 2019), we identify a single publicly available SR dataset–Reddit/MRS (Zhang et al., 2021). We supplement this by also evaluating on PERSONA-CHAT (Zhang et al., 2018), which similarly falls under the broader umbrella of open-domain dialogue. Below we provide further elaboration:

Reddit or MRS (Zhang et al., 2021) is, to the best of our knowledge, the only publicly available dataset created specifically for the SR setting. The dataset is multilingual, covering 10 languages and over $50M$ message-reply pairs extracted from the social-media site Reddit. As our focus is only on the monolingual setting, we use only the English portion of the corpus. Further, due to limited computational resources we train and evaluate on only a small subset of the data (randomly selected).

PERSONA-CHAT (Zhang et al., 2018) is a crowdworker-sourced dialogue dataset between pairs of speakers in which each speaker is assigned a brief persona comprising a few sentences, e.g. `I have a dog`. We simply concatenate this information to the message, following previous approaches (Humeau et al., 2020). As it is an open-domain dialogue dataset, it covers a broad range of possible conversations, and therefore provides another useful benchmark of performance for an SR system, which are often deployed in similarly open-domain environments.

5.3 Metrics

We use a weighted ROUGE (Lin, 2004) ensemble metric to evaluate performance, which is known to be well correlated with click-through rate in the SR setting (Zhang et al., 2021). This consists of a mixture of 1/2/3-grams for ROUGE-F1:

$$\frac{\text{ROUGE-1}}{6} + \frac{\text{ROUGE-2}}{3} + \frac{\text{ROUGE-3}}{2} \quad (6)$$

5.4 Hyperparameters

We train our models using the Adam optimizer (Kingma and Ba, 2014) for 3 epochs, with an initial learning rate of $5e-5$ and linear decay, and a batch size of 8. We truncate the message and response to the last 64 tokens each. We initialise our models from the DistilBERT checkpoint (Sanh et al., 2019),[2] which is a $66M$ parameter transformer trained via knowledge distillation on BERT. During inference, we set $K = 3$ which is a standard number for SR (Zhang et al., 2021). We also set the number of candidates initially retrieved by the Matching model $N = 15$, which previous work has shown provides a good trade-off between accuracy and latency (Deb et al., 2019). For SIMSR, we set the number of simulations $M = 25$. For both PERSONA-CHAT and Reddit we use the entire training set to retrieve from (i.e. Y_R). In early testing, we explored using heuristic techniques to create a more deduplicated candidate pool, but found limited benefit, and therefore opted for this simpler approach.

During deployment, although SR systems produce multiple replies, only *one* of them needs to be relevant. To replicate this, we only record the maximum ROUGE across the $K = 3$ replies outputted. We also report Self-ROUGE (Celikyilmaz et al., 2020), which is an unreferenced metric that measures the diversity of the predicted replies. For each reply $y_k \in Y_K$, we treat y_k as the prediction and the other two replies as the references, using the same ROUGE metric as above. Note that a lower Self-ROUGE indicates *more* diversity.

5.5 Choosing a Search Strategy

Table 1 shows the performance of SIMSR under different search strategies. This is motivated by two sub-questions: (1) how robust is SIMSR to the choice of search strategy? (2) What trade-offs are involved between relevance, diversity and efficiency?

Exhaustive search unsurprisingly performs the best both in terms of relevance and diversity, but is the least efficient and would not scale to larger values of N. More interesting is the trade-off between relevance and diversity that occurs between the Ablative and Greedy methods. Greedy performs slightly better in relevance, perhaps suggesting that the longer sequences involved in the Ablative method leave more opportunity for errors to be propagated. However, Greedy performs significantly worse in diversity. While a high diversity is not always a good thing (e.g. random guessing would also have a high diversity), Ablative's diversity is much closer to that obtained by Exhaustive search. Sample and Rank consistently gave the worst results, suggesting randomly constructing tuples is insufficient for finding high-value tuples.

Overall, these results show that SIMSR is reasonably robust to the choice of search strategy. Going forward, we opt to use Ablative search for subsequent experiments which provided arguably the best trade-off in terms of relevance, diversity and efficiency by a small margin.

5.6 Main Results

Table 3A-B summarises our main results. Across both tasks, we find that additional filtering/diversification measures improve the diversity of the suggested replies, but provide only limited improvement to relevancy. We argue this reflects the fact the these methods often involve trading off relevance for diversity, such as MMR, which explicitly scores replies as a linear combination of their relevancy to the message and their similarity to other replies in the reply set. Similarly, whilst the out-of-the-box Topic classifier sometimes produced outputs that were more diverse than the other baselines, this came at the cost of reduced relevance, due to it being too coarse-grained–i.e. often a given message required multiple replies from the *same* topic.

Contrastingly, we show our method is able to consistently improve on both relevancy and diversity for both tasks. On Reddit, relevancy improves by up to 14% and diversity by up to 21%; on PERSONA-CHAT, relevancy improves by 18% and diversity improves by 6%. All results are statistically significant on a t-test with p-value < 0.01. The main difference between the datasets is that

[2] https://huggingface.co/distilbert-base-uncased

Section	Method	Reddit		PERSONA-CHAT	
		ROUGE ↑	Self-ROUGE ↓	ROUGE ↑	Self-ROUGE ↓
(A) Baselines	Matching	2.04	6.92	6.61	12.44
	Matching + Topic	2.01	3.17	6.42	11.77
	Matching + MMR	2.17	5.19	6.66	10.76
	MCVAE	2.12	3.99	6.52	8.93
(B) Our Method	SIMSR	**2.40**	**2.36**	**7.71**	**8.39**
(C) Ablations	- Multi-reply	2.02	19.77	7.03	35.24
	- Simulation	2.04	6.92	6.61	12.44

Table 3: Performance of SIMSR (B) compared to baseline approaches (A) and ablations (C) on the Reddit and PERSONA-CHAT Test sets. All results are statistically significant on t-test with p-value < 0.01.

PERSONA-CHAT is a less noisy dataset, being made by crowdworkers, and therefore both metrics are comparatively higher.

5.7 Ablations

We consider the question of whether SIMSR is simply learning to predict individual replies that have a high expected score, rather than learning to take advantage of interdependencies between replies. To this end, in Table 3C we present an ablation ('- Multi-Reply') that selects the top-K replies according to their *individual* scores in simulation, without considering their scores at the *tuple*-level, i.e. $\text{TopK}(\{E[h(y, y_n)]\}_{n=1}^{N})$. We also present a version without simulation at all as a baseline comparison, which is equivalent to the Matching model in Table 3A.

Results show that removing multi-reply significantly harms performance. Versus the baseline, there is no improvement on Reddit, while there are only limited gains on PERSONA-CHAT, suggesting most of the performance gains from SIMSR are due to the ability to account for interdependencies within the reply set. We hypothesise the reason for the difference between the two datasets is because PERSONA-CHAT is a less noisy dataset, and therefore selecting individual replies with a high expected similarity may provide some benefit. Diversity is especially harmed, and even is significantly less diverse than the baseline. This is unsurprising, given maximising the similarity of each reply to the same set of simulated replies implicitly encourages responses to be similar.

5.8 Case Study

Table 4 presents two case studies comparing the qualitative performance of SIMSR versus a selection of baseline methods. In both case studies we see SIMSR is able to represent three diverse intents across its predictions versus only one or two intents for the Matching and MMR models. In the left example, SIMSR is crucially able to capture including both a positive and a negative intent, unlike the baselines. In the right example, SIMSR successfully avoids duplicating the `I'm glad` intent. Note that in both cases it would be impractical to use heuristic measures to deduplicate the intents (e.g. removing replies with only 1 word edit distance) as there is often only partial term-level overlap between the utterances.

5.9 Latency

Table 5 validates the limited latency impact of SIMSR compared to the baseline methods. We used an NVIDIA GeForce RTX 3060 Ti GPU and CPU operations were conducted by an AMD Ryzen 7 5700G with Radeon Graphics. For the initial retrieval, we pre-compute the reply embeddings and store them in a FAISS index (Johnson et al., 2017). Overall, we find SIMSR is able to maintain comparable latency to other methods which encorporate post-hoc diversification methods such as MCVAE and MMR. The small latency difference for SIMSR is mainly due to the iterative search and evaluation process not using any low-level optimisation in the code or multiprocessing. Topic is the slowest due to the additional inference cost of the Topic classifier.

6 Conclusion

In this work, we have presented a method for generating sets of replies for Smart Reply systems, using model-based simulation and a range of search strategies to discover high-value reply sets, without the need for any additional training. Our method outperforms existing SoTA methods on both datasets tested, and we have supported our results by detailed analysis of the effect of different search strategies, demonstration of the impor-

PERSONA-CHAT	Reddit
Message: *So do you have any pets?*	**Message:** *where? i've always wanted to be in one!*
Matching	
No, no pets. Do you have any	I'm so glad I'm not the only one.
No, no pets. You?	glad i'm not the only one....
No, I do not have any pets. What are some things you like	Wait... They said I'll be the the first...
MMR	
I do not have any but I do want a dog	I will have one of everything, please.
No, no pets. You?	I'm so glad I'm not the only one.
No, no pets. Do you have any?	glad i'm not the only one....
SIMSR	
No, I do not have any pets.	I'll be there, too. Also my first time seeing them. Can't wait.
Nope no pets at the moment. How are you?	Glad I wasn't the only one
Yes I have 2 dogs.	ME TOO. We need to go find one.

Table 4: Examples of model outputs on the PERSONA-CHAT (left) and Reddit (right) Test sets. SIMSR produces replies that capture multiple possible user intents, while the other approaches capture a more limited range of intents.

Method	Latency (ms)
Matching	23.3
Matching + Topic	45.5
Matching + MMR	24.5
MCVAE	25.9
SIMSR	29.9

Table 5: Latency of SIMSR compared to baseline approaches on the Reddit Validation set.

tance of accounting for interdependencies between replies, and a detailed case study. Future work could consider whether it is possible to improve the quality of the initial retrieval (e.g. by training on sets of replies), or other methods for scoring response similarity during simulation.

Acknowledgements

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

Limitations

While our approach is able to optimise over the retrieved shortlist of replies, it does not improve the initial retrieval from the candidate pool, which still scores individual candidates, rather than reply sets, using the Matching model. This is a limitation that is shared with prior baseline methods. A further limitation is that we only consider the monolingual setting, whereas many deployed SR applications have an international footprint. Learning a multilingual Matching model in SR is known to have additional challenges (Deb et al., 2021). Another limitation is that our model is only tested on public dialogue datasets, due to actual conversations on platforms using SR being proprietary. Therefore, while our techniques should work well in the instant messaging setting, our methods have not been directly tested in the email setting.

Ethical Considerations

As neural dialogue models have grown in expressive capabilities and fluency, ethical considerations are an increasingly prominent issue. Key considerations typically centre around model's tendencies (1) to produce information that is factually inaccurate (Shuster et al., 2021) or (2) to repeat toxic/biased behaviour from the training data (Xu et al., 2020). Compared to vanilla dialogue models, these risks are mitigated in SR: (1) SR is usually limited to short-form replies that express simple information, and is therefore less likely to lead to the kinds of hallucination seen in longer-form answers; (2) SR typically does not generate tokens sequentially, but retrieves responses from a pool of candidates, which can be vetted in advance. Note however, this does not prevent replies that are contextually inappropriate when paired with a particular message, e.g. `Do you hate people?` → `Yes, I do`. The human-in-the-loop, who must ultimately choose and be accountable for whether or not to select one of the suggested replies, can be seen as a risk mitigant compared to vanilla chatbots.

Conversely however, Wenker (2023) identify risks pertaining to a loss of human agency, such as due to a user selecting a sub-optimal reply to save time or being primed by the replies. This could lead to people being more trusting of an SR-generated reply versus receiving a reply from a chatbot, due to the belief that a human ultimately is behind it. We also only experimented with datasets that were released by previous studies, which are publicly available. These datasets (especially Reddit) often contain toxic/biased behaviour which developers should bear in mind if using this system in a deployment context.

References

Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Vitor Silva, Leonardo Neves, and Francesco Barbieri. 2022. Twitter topic classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3386–3400, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jaime G. Carbonell and Jade Goldstein-Stewart. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *ArXiv*, abs/2006.14799.

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.

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*.

Budhaditya Deb, Guoqing Zheng, Milad Shokouhi, and Ahmed Hassan Awadallah. 2021. A conditional generative matching model for multi-lingual reply suggestion. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1553–1568, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel De Freitas, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot. *ArXiv*, abs/2001.09977.

Matthew 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. *ArXiv*, abs/1705.00652.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Youngsoo Jang, Seokin Seo, Jongmin Lee, and Kee-Eung Kim. 2021. Monte-carlo planning and learning with language action value estimates. In *International Conference on Learning Representations*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7:535–547.

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*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016a. Deep reinforcement learning for dialogue generation. In *Conference on Empirical Methods in Natural Language Processing*.

Xiujun Li, Zachary Chase Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung (Vivian) Chen. 2016b. A user simulator for task-completion dialogues. *ArXiv*, abs/1612.05688.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Kseniia Palin, Anna Maria Feit, Sunjun Kim, Per Ola Kristensson, and Antti Oulasvirta. 2019. How do people type on mobile devices?: Observations from a study with 37,000 volunteers. *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*.

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, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Conference on Empirical Methods in Natural Language Processing*.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, L. Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *ArXiv*, abs/1712.01815.

Richard S. Sutton and Andrew G. Barto. 2005. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 16:285–286.

Benjamin Towle and Ke Zhou. 2022. Learn what is possible, then choose what is best: Disentangling one-to-many relations in language through text-based games. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4955–4965, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Paul Varcholik, Joseph J. LaViola, and C. Hughes. 2012. Establishing a baseline for text entry for a multi-touch virtual keyboard. *Int. J. Hum. Comput. Stud.*, 70:657–672.

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.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *ArXiv*, abs/2010.07079.

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.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur D. Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Annual Meeting of the Association for Computational Linguistics*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*.

A Artifacts: code, datasets and models

This section lists the licences for the code, datasets and models used in the paper ('Artifacts'): DistilBERT (Sanh et al., 2019) is under Apache-2.0 licence; PERSONA-CHAT (Zhang et al., 2018) is under CC BY 4.0; The topic classifier (Antypas et al., 2022) is fine-tuned from the pre-trained transformer RoBERTa (Liu et al., 2019) which is under the MIT licence; The Reddit dataset (Zhang et al., 2021) is available under the MIT licence; Our code pertaining to this paper is released under the MIT licence.

B Experiment details

Models were trained using an NVIDIA GeForce RTX 3060 Ti. Training took no longer than an hour for any one model, as they were fine-tuned from pre-existing pre-trained models and the datasets were comparably small. Hyperparameters were selected using using recommended values for fine-tuning (Devlin et al., 2019), and where not explicitly specified use default values from the HuggingFace Trainer class. Experiments were run using a single random seed. For evaluation, ROUGE was calculated using the rouge-score Python package [3].

C Further examples

Table 6 shows further examples of SIMSR's predictions versus the other baselines.

[3] https://pypi.org/project/rouge-score/

	PERSONA-CHAT
Message:	*i do, i turn up ed sheeran on my ipod and go to my favorite waterfall.*
Matching	that is nice do you like to hike ? do you like to hike ? that sounds like fun . do you have a favorite artist ?
MMR	who is your favorite artist ? that is nice do you like to hike ? do you like to hike ?
SIMSR	do you like to hike ? that is amazing . i love nature . who is your favorite artist ?
	Reddit
Message:	*deal. i'm in newcastle haha*
Matching	See you there! Great, see you there! I'm In
MMR	Where? I'm low on gas and you need a jacket. See you there! Great, see you there!
SIMSR	see you in 15 minutes. Yeah sure Sounds good to me tho

Table 6: Additional examples of model outputs on the PERSONA-CHAT (top) and Reddit (bottom) Test sets.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☑ A2. Did you discuss any potential risks of your work?
Ethical Considerations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

5.1,5.2,5.4

☑ B1. Did you cite the creators of artifacts you used?
5.1,5.2,5.4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix A

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix A

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
5.2

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
5.2

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
5.4, Appendix B

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
5.4, Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix B

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.