Debiasing large language models for persona-based dialogue systems

Anonymous ACL submission

Abstract

Persona-based chatbots are conversational AI systems designed to emulate the behaviour and characteristics of specific personas, whether from real life or fiction. Previous research has mainly concentrated on aligning chatbot responses with predefined personas. However, manually creating these personas can be time-consuming and may not fully capture all aspects of an individual's personality. This study introduces a new task: *persona generation*, aiming to generate diverse and high-quality personas before or during conversations. Inspired by the success of large language models, we use ChatGPT to accomplish the task and observe that the model has a strong sampling bias towards generating personas resembling a specific demographic group. To increase persona diversity, we propose two strategies: (1) Chain-of-decision prompting and (2) Listing sampling. Experimental results show that our approaches significantly outperform temperature sampling and logit suppression in terms of diversity. As our method is task-agnostic and does not necessitate additional training, it can be applied to various tasks that are susceptible to bias from large language models.

1 Introduction

Creating an open-domain chatbot capable of engaging in natural and unrestricted conversations has long been a goal in the field of natural language processing. The challenge has been the wide range of potential conversation topics and the absence of a clearly defined objective (Roller et al., 2020). However, recent advancements in large language models (LLMs), such as ChatGPT and GPT4 (OpenAI, 2023), have brought significant progress (Lee et al., 2023; Lu et al., 2023). These models have shown remarkable success in responding to a variety of novel questions.

Traditionally, chatbots have been used as virtual assistants for casual chat and information retrieval. As most of them lack human-like attributes such as personalities, emotions, and opinions, the interactions tend to be monotonous and uninteresting. There has been a growing interest in developing chatbot applications based on personas, where chatbots assume the identity of specific individuals (Zhang et al., 2018; Xu et al., 2021; Ahn et al., 2023). Successful examples of such applications include Replika (Replika) and Character AI (CharacterAI). Persona-based chatbots offer users a genuinely authentic and personalised experience, fostering long-term rapport. Users can also be able to switch between different chatbots if they find one unengaging.

Most research on persona-based chatbots assumes that the personas are predefined, and the objective is to generate responses from the bot that align with those personas (Zhang et al., 2018; Xu et al., 2021; Han et al., 2022). However, crafting personas manually is a time-consuming process, involving a wealth of information, such as demographic backgrounds, personalities, opinions, and goals. Previous attempts have been made, but they often resulted in personas limited to a few brief sentences, which may not capture the depth required (Zhang et al., 2018; Mazaré et al., 2018). As human-generated personas have scalability limitations, it is common that one manually-created chatbot is shared by multiple users. This potentially discourages those seeking dedicated chatbots for their individual use.

We explore the research question: *How can we automatically create a large number of personas, each with a distinct background and personality?*. These personas can be generated in two ways: first, *In-advance generation*, where a separate model creates a detailed persona that is then used to guide the chatbot's responses, and second, *On-the-fly generation*, where the chatbot itself acts as the persona generation model, gradually revealing the persona through its interactions. Automatic persona generation not only provides a unique and authentic

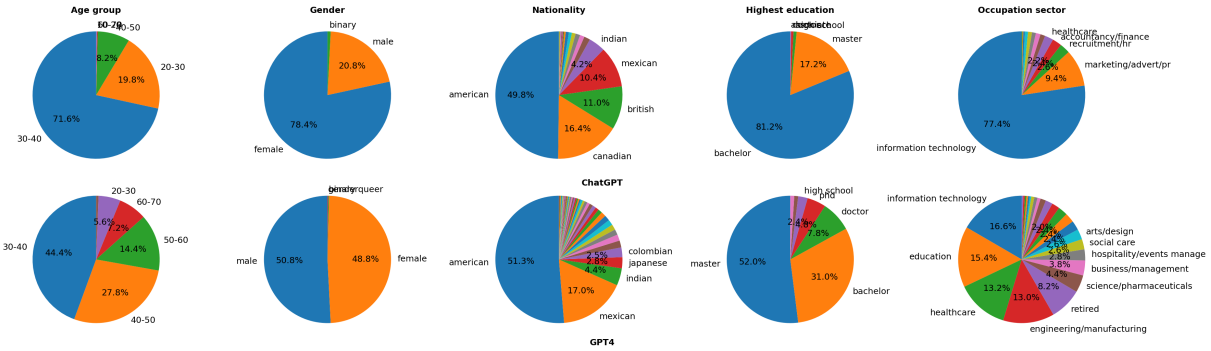

Figure 1: Demographic distributions of 500 personas generated by ChatGPT and GPT4 with default parameters. Prompt details can be found in Section 3.1.1

experience for chatbot users but also has significant potential in various applications. These include human behaviour simulation (Park et al., 2023), creating non-player characters in video games, or constructing persona-based dialogue datasets (Cao et al., 2022; Mazaré et al., 2018).

We use LLMs for persona generation thanks to their impressive performance in handling arbitrary tasks. As shown in Figure 1, we have noticed a significant sampling bias in the personas generated by ChatGPT, primarily favouring a specific group: middle-aged Americans working in the technology industry. Although GPT4 can significantly improve persona diversity, the sampling bias remains strong for certain categories such as nationality and education. Despite various proposed methods to mitigate bias (Gallegos et al., 2023), many of them require full control over the system, making them unsuitable for black-box models like ChatGPT. We propose two prompting strategies to debias LLMs and enhance text diversity:

- **Chain-of-decision (COD) prompting:** This method involves breaking down the prompt task into a series of decisions. The LLM suggests different choices for each decision, from which one is selected. The final output is generated based on all the selected choices. This approach is ideal for generating long texts, such as In-advance persona generation.

- **Listing sampling:** The LLMs is prompted to provide a list of diverse responses for the prompt task. From this list, one response is randomly chosen as the final output. We suggest this approach for generating short texts like On-the-fly persona generation.

Experiments show the superiority of our proposed methods over task-agnostic techniques like temperature sampling and logit suppression. We achieve remarkable improvements in persona diversity with very little degradation in persona coherence. As our approach is versatile and does not necessitate further training, it can be adapted to mitigate bias in various text generation tasks.

2 Related works

2.1 Debiasing large language models

Large language models have become the standard for text generation with impressive zero-shot capabilities across various tasks (OpenAI, 2023; Touvron et al., 2023; Anil et al., 2023). Their success can be attributed to the significant increase in model parameters and training data. However, due to the unverified and unchecked nature of most of the data, LLMs may unintentionally perpetuate stereotypes, misrepresentations, and discriminatory decisions, leading to adverse social consequences (Gallegos et al., 2023). Several previous studies have focused on addressing the issue of stereotyping bias in these models, where the generated text reinforces stereotypes related to gender, race, ethnicity, religion, and other social factors (Gallegos et al., 2023). For instance, LLMs produce text expressing positive attitudes towards developed countries while displaying negative attitudes towards developing ones (Venkit et al., 2023).

This study investigates the bias related to overrepresentation, where generated content tends to favour specific groups while neglecting the contributions of marginalised communities. This bias arises from imbalanced labels of different demographic groups in the training data. Early methods of bias mitigation focus on improving the training data. These techniques seek to create more representative training data by generating

new underrepresented examples (Zmigrod et al., 2019; Qian et al., 2022) or upsampling existing representative/low-bias examples (Garimella et al., 2022; Han et al., 2021), or even generating an entirely new dataset with curated exemplary examples (Solaiman and Dennison, 2021). Another direction for bias mitigation is to alter the training loss function (Woo et al., 2023), updating next-word probabilities during training (Garimella et al., 2021), or removing specific neurons that contribute to harmful outputs (Webster et al., 2020). However, these methods require retraining or fine-tuning LLMs, which can be expensive and time-consuming.

Post-editing approaches, such as modifying the decoding process, offer better alternatives as they do not require additional training. One method involves using n-gram blocking to prevent the generation of potentially biased tokens or phrases during decoding (Gehman et al., 2020). Constrained beam search, as demonstrated in (Saunders et al., 2021), can create more gender-diverse texts. To exert more control during decoding, (Lu et al., 2020) propose using predicate logic statements to mandate the inclusion or exclusion of specific tokens. Others employ classification-based methods to identify harmful tokens or measure the negativity in candidate outputs (Dathathri et al., 2019; Schramowski et al., 2022). These measurements can then be utilised for filtering or ranking candidates, as shown in (Shuster et al., 2022).

Our research aligns closely with the approach of optimising prompts to minimise bias. This involves modifying prompts to encourage fairness in the generated outputs. For instance, (Mattern et al., 2022) explores prompting languages to reduce gender bias related to occupations. (Abid et al., 2021) and (Venkit et al., 2023) aim to diminish negativity bias toward specific groups by adding positive phrases to the original prompts. Similarly, (Sheng et al., 2020) identifies triggers that promote positivity for social groups. These methods primarily address stereotype issues. In contrast, our paper focuses on optimising prompts not only to address sampling bias but also to increase text diversity.

2.2 Diverse text generation

Our research work is also closely related to diverse text generation, where the goal is to generate varied and diverse outputs while maintaining coherence. This is still ongoing research despite the success of LLMs (Padmakumar and He, 2023). Several approaches have been proposed to promote diversity by changing the predominant maximum likelihood training objective. (Shao et al., 2019) propose a planning-based hierarchical variational model for generating long and diverse texts. (Du et al., 2022) and (Bao et al., 2020) introduce a latent structured variable model, and (Su et al., 2022) propose a constraint learning framework for similar purposes. However, these methods necessitate significant alterations to the training process, which may not be ideal for users seeking to use pre-trained LLMs.

A more viable solution is to improve decoding algorithms. For instance, (Vijayakumar et al., 2016) propose diverse beam search to decode a list of diverse outputs by optimising for a diversity-augmented objective. Instead of always selecting the most probable tokens, techniques like Nucleus sampling (Holtzman et al., 2019) and Top-K sampling (Fan et al., 2018) advocate choosing less likely tokens to enhance diversity. Additionally, methods like logit suppression (Chung et al., 2023) discourage the generation of overly common tokens. (Su and Collier, 2022) demonstrate that contrastive search decoding outperforms previous methods in terms of both diversity and coherence. Unlike existing techniques, our method only requires changes in the prompting language, without any modifications to the generation model. This makes it adaptable for any off-the-shelf LLMs.

3 Methodology

3.1 LLMs for persona generation

3.1.1 In-advance generation

In this setting, the chatbot's persona is created separately in advance and then used to condition the chatbot's responses throughout the conversation.

To generate N different personas, we prompt LLMs N times as follows:

Create a random individual profile that includes demographic information. Please consider demographic diversity when generating the profile.

We explicitly use the term *random* and an instruction from the second sentence to encourage the randomness and fairness. We also ask LLMs to include demographic information for later evaluation, but this should not prevent the model from generating other information about the persona.

3.1.2 On-the-fly generation

It is possible for users to enquire about personal information that is not included in the chatbot's pre-

defined persona. In such cases, the On-the-fly setting is designed to generate new information about the chatbot during the conversation, assuming that the pre-defined persona is either incomplete or unavailable. This process is carried out together with the response generation task, where the chatbot's persona is revealed in the generated response. We prompt LLMs for response generation as follows:
Given this conversation:
...
Person A:
Person B:
Person A:
Imagine you are person B and act as if you were a real individual who willing to disclose everything. Please compose the next response for person B in no more than two sentences.

To obtain N different personas, we conduct N different conversations, where *Person B* assumes the role of the chatbot, and its persona is extracted for assessments. *Person A* acts as a persona seeker, posing questions such as *What do you do for work?* to extract persona information from *Person B*. *Person A* can be a human or another chatbot. In this paper, we use ChatGPT to simulate *Person A*. Prompt details can be found in Appendix A.1.2. Each conversation begins with two pre-defined greeting turns, which are then followed by automatic responses from *Person A* and *Person B*.

3.2 Chain-of-decision prompting

It could be argued that for LLMs to accomplish any given task, there are always several decisions to be made implicitly during the generation. This is different from what humans do in writing, where the key decisions and content structures are decided explicitly beforehand. For example, to write a poem, the author has to decide on the topic, the structure, the tone, and so on, before they start composing.

We propose Chain-of-decision (COD) prompting, where we ask the LLMs to do the planning before generating the content. The five core steps are as follows:

1. **Decision listing:** Given the task description, LLMs generates a series of key decisions $D = \{D_1, D_2, ..., D_n\}$ needed to accomplish the task.

2. **Choice listing:** For each decision D_i, LLMs generates a diverse list of possible choices $C_i = \{C_i^1, C_i^2, ..., C_i^m\}$ for that decision.

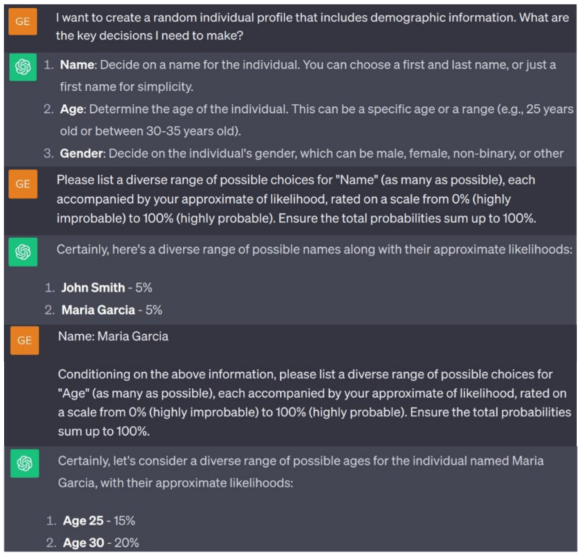

Figure 2: An example of COD prompting for In-advance persona generation

3. **(optional) Choice likelihood estimation:** For each choice C_i^j, LLMs estimates its likelihood from 0% to 100%. The sum probability of the entire list C_i should be 100%.

4. **Choice selection:** A choice C_i^* is selected from the list C_i, either randomly or based on likelihood from step 3. The selection is not done by LLMs but rather by an external package (e.g., Python random library).

5. **Content generation:** The selected choices are concatenated together with the original task description. LLMs then generates the final output using the concatenated prompt.

To ensure all the choices align and make sense together, the process is done sequentially and dependently. This means that the choice listing and choice likelihood estimation of decision D_i are conditioned on the previously selected choices $C_{k<i}^*$. This is done by concatenating these selected choices into the input prompt when performing steps 2 and 3. Please see Appendix A.1.1 for more details about the prompt template. Figure 2 shows an example of how COD is executed.

By informing LLMs about the different choices it can make, we believe this will lead to an increase in diversity and a reduction in bias in the generated text. Since COD requires iterative prompting, we suggest using this method for long text generation, such as In-advance persona generation.

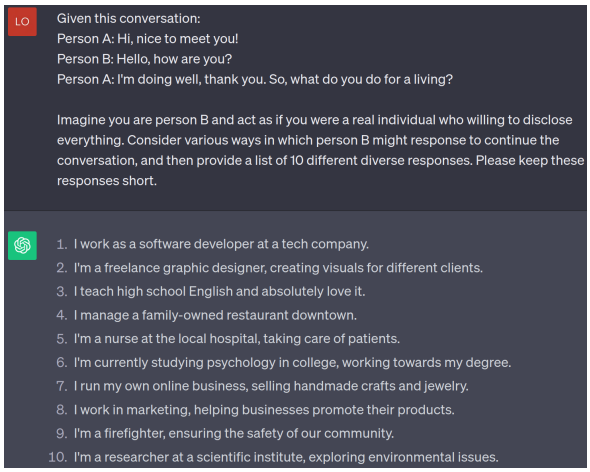

Figure 3: An example of Listing sampling for response/persona generation

3.3 Listing sampling

Although COD is task-agnostic, its sequential nature might significantly increase the latency of the generation. This makes it less suitable for short text generation or applications that require very fast response such as chatbots. Therefore, we propose Listing sampling with LLMs for diverse short text generation as the following template:
I want to <task description>. List a diverse set of <N> possible responses:

After creating a list of N potential responses, we randomly choose one to be the final answer. It is important to note that the generation of each candidate is influenced by previous generated candidates, ensuring both diversity and coherence. This offers a significant advantage over other sampling techniques like beam search or nucleus sampling as these methods generate candidates simultaneously. As a result, while the candidates vary in wording, their semantic meanings may remain similar.

We apply Listing sampling for On-the-fly persona generation as shown in Figure 3.

4 Experiment settings

4.1 Baselines

Due to cost-related reasons, we choose ChatGPT (gpt-3.5-turbo-0613) as the LLMs for experiments. The model is released by OpenAI for commercial use[1]. Based on the available options of ChatGPT API[2], we experiment with two strategies to improve persona diversity namely temperature sampling and logit suppression. These two approaches are also task-agnostic when it comes to debiasing LLMs, requiring no specific customisation or expertise.

4.1.1 Temperature sampling

Temperature sampling is a decoding technique for controlling the randomness of the generated text (Ficler and Goldberg, 2017; Fan et al., 2018). The temperature parameter adjusts the distribution of the predicted probabilities. Higher temperatures make the distribution flatter, encouraging the model to generate more diverse outputs. Lower temperatures sharpen the distribution, making the model more deterministic, generating more conservative outputs. As ChatGPT API offers a temperature range from 0 to 2, we experiment with two values: *temperature=1.0* (default) and *temperature=1.3*. We found that increasing the temperature further leads to irrelevant and out-of-vocabulary words, similar to (Chung et al., 2023). In combination with higher temperatures setting, we can request the API to provide alternative responses, which are ranked lower in terms of likelihood. We can then choose one of these alternatives randomly to enhance the diversity of the final response.

4.1.2 Logit suppression

Another method to enhance diversity is through logit suppression, applicable when generating a diverse collection of documents (Chung et al., 2023). This approach involves tracking the frequency of tokens generated so far. The generation of the current document depends on previously generated documents, where the probabilities of highly frequent tokens are suppressed.

The current ChatGPT API includes a feature called the logit bias parameter, which allows users to control how individual tokens are prioritised or de-emphasised. This parameter accepts values ranging from -100 to 100, where negative values decrease the likelihood of selection, and positive values increase it. We first identify the 300 most frequently occurring tokens. For each of these tokens, we calculated its appearance ratio, defined as the number of documents containing the token divided by the total number of documents. We then multiply a bias weight of -7.5 to the appearance ratio. As certain punctuation marks are necessary to produce accurate texts, we only suppress tokens that are at least 2 characters long. Additionally, we explore the suppression of only demographic tokens such as gender-related (i.e. *Female*) or nationality-related tokens (i.e. *American*). This adds more

[1] https://openai.com/policies/terms-of-use
[2] https://platform.openai.com/docs/guides/gpt

relevant tokens to the top 300. Suppressing only demographic tokens also reduces the risk of degenerating common but important tokens such as stopwords or function words. As a result, we can further increase the bias weight from 7.5 to -15.0 to improve persona diversity. The extraction of demographic tokens is described in Section 4.2.3 and Appendix A.1.3.

4.2 Evaluation metrics

Assume we generate a set of N personas, denoted as $P = \{P_1, P_2, ..., P_n\}$. Each persona P_i contains a set of attribute values $A_i = \{a_i^1, a_i^2, ..., a_i^m\}$, where a_i^j represents a particular attribute value (such as *female*) corresponding to the j-th attribute (such as *gender*). Let $A^j = \{a_1^j, a_2^j, ..., a_n^j\}$ be a collection of all values of the j-th attribute, extracted from P.

4.2.1 Diversity score

To measure the diversity of P, we can calculate the diversity score for each attribute and then take the average. Shannon entropy can be applied to measure the randomness/uncertainty score of the j-th attribute as follows:

$$H(A^j) = -\sum_{k}^{K} P(a_k^j) log(P(a_k^j))$$

Where $H(A^j)$ represents the entropy of A^j, a_k^j represents each possible value of A^j, $P(a_k^j)$ represents the appearance ratio of the value a_k^j, and K is the number of distinct values of A^j. As can be seen, the increase in the number of distinct values or the decrease in the appearance ratio of each value will lead to a higher diversity score.

We also report other metrics: Top-2 percentile and Coverage. Top-2 percentile refers to the sum of the appearance ratios of the two most common values in A^j. Coverage is calculated as follows:

$$Coverage(A^j) = \frac{K}{upper\_bound(K)}$$

Where $upper\_bound(K)$ represents the maximum number of unique values that can exist in A^j. Please see Appendix A.2 for more details.

4.2.2 Coherence score

We also need to make sure that for each generated persona, the attributes are all aligned and make sense together. We prompt GPT4 to rate the coherence score for each persona as follows:

Given this profile:
<Attribute name #1>: <Attribute value #1>
...
<Attribute name #M>: <Attribute value #M>
Check this profile to see if all the details are aligned and make sense together. Then rate the profile from scale 1 (very incoherent) to 5 (very coherent) without any explanation.

4.2.3 Attribute extraction

This paper only focus on evaluating specific attributes: age group, gender, nationality, occupation sector, and highest education level. We have defined a list of possible values for each attribute. For instance, age groups are categorised as {0-10, 10-20,...70+}. By prompting ChatGPT, we can extract age, gender, nationality, occupation, and education from the generated personas/dialogues. Since the extracted data can be noisy and varied, we standardise them by mapping each value to the predefined values mentioned earlier. To achieve this, we employ a set of heuristics and also prompt ChatGPT for assistance. See Appendix A.1.3 for details.

5 Experiment results

We generate 500 personas with each of the approaches mentioned in Section 3. For On-the-fly setting, we experiment with (randomly) choosing other lower-ranked candidates as the final response. For In-advance setting, only the highest-scored candidate is considered. The results are visualised and reported in Figure 4, 5, and Table 1.

Default setting As shown in Figure 4, when using the default temperature setting (*temp=1.0*), there is a noticeable sampling bias in ChatGPT's outputs. Specifically, the bias is towards a specific demographic: American males aged 20-40 who work in the information technology industry with a bachelor's degree. When we aggregate the percentages of the two most common values for each persona attribute and calculate the average, the resulting figure is 92.5%. This implies that a relatively small group of people represents almost the entire population. The sampling bias is reduced when personas are generated in advance, highlighting the advantage of persona-based dialogues over persona-free ones. However, even with in-advance persona generation, the figure for the Top-2 percentile remains high at 88.4%.

Higher temperature Increasing the temperature from 1.0 to 1.3 results in improved diversity for

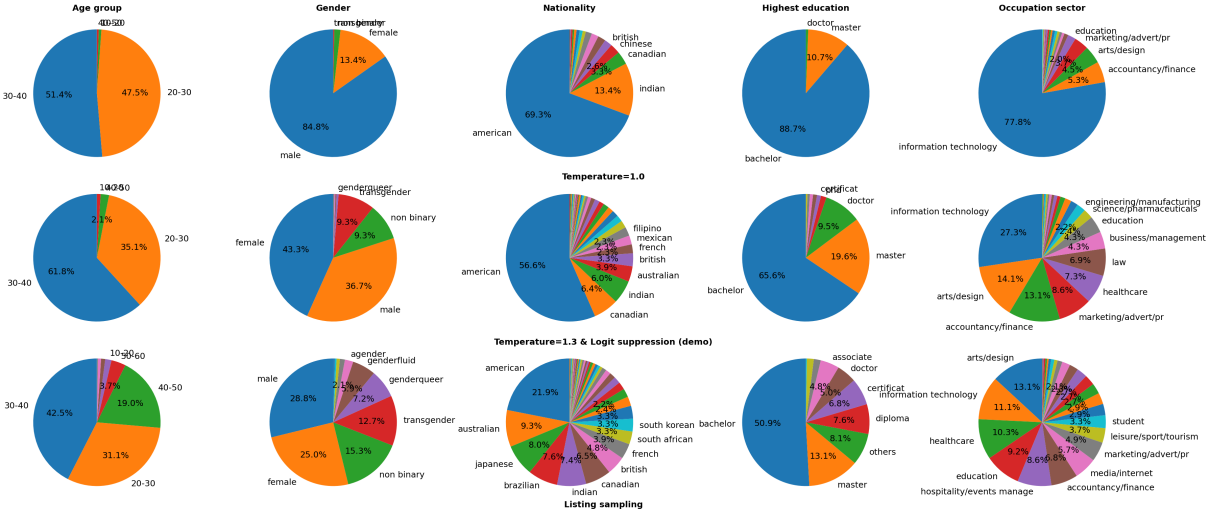

Figure 4: Demographic distributions with On-the-fly persona generation

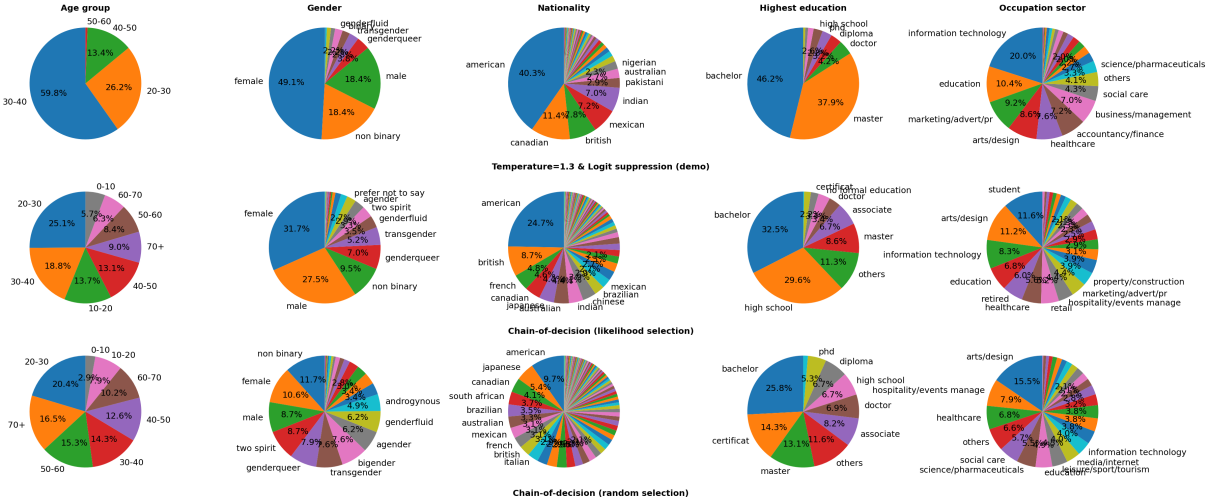

Figure 5: Demographic distributions with In-advance persona generation

both generation settings without a decrease in coherence score. The most significant enhancement is observed in the In-advance setting for the occupation attribute, showing a 75% relative improvement in diversity. Smaller improvements were also noted in other persona attributes. We also observe that increasing the number of candidate outputs and then randomly selecting one as the final response does not yield significant improvements.

Logit suppression Similarly, reducing the occurrence of frequent tokens mitigates the sampling bias of ChatGPT, as demonstrated by a 15% and 18% decrease in the Top-2 percentage for In-advance and On-the-fly generation, respectively. The most substantial improvements are observed in the occupation attribute, while the least improvements are seen in the age group attribute. One of the limitations of logit suppression is that it can only suppress tokens with an exact match, not tokens with different lexical forms but belonging to the same category. For example, *24* and *25* are lexically different but belong to the same *20-30* age group. Lastly, suppressing only demographic tokens is significantly better compared to degenerating all tokens; however, it compromises the task-agnostic nature of the approach.

Listing sampling remarkably outperforms the combination of higher temperature and logit suppression, reducing the Top-2 percentile from 73.3% to 49.4% and increasing diversity from 2.15 to 3.04. However, the approach still exhibits sampling bias towards age and education attributes. One solution is to increase the number of different responses N to enhance diversity further. This, of course, might come at the cost of increased latency and a slight degradation of the coherence score, as shown in Ta-

Methods	Top-2 percentile	Shannon entropy						Coherence
		Age	Gen	Nat	Edu	Occ	Avg	
On-the-fly persona generation								
-Temp=1.0, 1-best candidate	92.5%	1.09	0.71	1.78	0.54	1.51	1.13	5.00
-Temp=1.3, 1-best candidate	88.8%	1.07	0.96	2.24	0.73	1.88	1.38	5.00
-Temp=1.3, 10 candidates	88.3%	1.06	1.04	2.29	0.71	2.02	1.42	5.00
+ Logit suppression (demo)	73.3%	1.14	1.80	2.83	1.54	3.42	2.15	5.00
-Listing sampling (N=10)	49.4%	**1.92**	2.65	4.18	2.37	4.07	3.04	4.97
+ Logit suppression (demo)	**47.9%**	1.88	**2.84**	**4.35**	**2.53**	**4.11**	**3.14**	4.96
In-advance persona generation								
-Temp=1.0	88.4%	1.14	0.80	2.38	0.80	1.40	1.30	5.00
-Temp=1.3	82.0%	1.36	0.99	2.58	1.06	2.45	1.69	5.00
+ Logit suppression (all)	74.3%	1.31	1.41	2.95	1.40	3.65	2.14	5.00
+ Logit suppression (demo)	64.0%	1.38	2.17	3.46	1.88	3.91	2.56	5.00
-Chain-of-decision (likelihood)	44.3%	2.83	2.93	4.67	2.56	4.35	3.47	4.91
-Chain-of-decision (random)	**27.6%**	**2.86**	**4.02**	**5.66**	**3.04**	**4.35**	**3.99**	4.88

Table 1: Persona generation results. *Age, Gen, Nat, Edu, Occ, and Avg* refer to the age group, gender, nationality, highest education, occupation sector, and average, respectively. *Temp=1.0, 1-best candidate* refers to temperature sampling at 1.0, where the best-scored candidate is the final response. *Temp=1.3, 10 candidates* refers to temperature sampling at 1.3, where 10 candidates are returned by the API, and one is randomly selected in the final response. *Logit suppression (all/demo)* refers to whether all tokens or only demographic tokens are considered for suppression

Methods	Coverage rate(%)				
	Age	Gen	Nat	Edu	Occ
Temp=1.0	62.5	5.17	9.69	35.7	46.4
Temp=1.3 +Logit	50.0	18.9	22.9	64.2	92.8
COD	**100**	**43.1**	**45.9**	**71.4**	**96.4**

Table 2: Coverage rate of different approaches for In-advance persona generation

ble 1. Although Listing sampling can be combined with logit suppression to enhance diversity, the approach itself does not require keeping track of frequent tokens, simplifying the generation pipeline.

Chain-of-decision achieves the best results with a diversity score of 3.99, along with the lowest Top-2 percentile at 27.6%. By instructing ChatGPT to explicitly list possible choices for each decision, we can identify rare or uncommon options when generating the persona. This is shown in Table 2 with remarkably high coverage rates of 100%, 71.4%, and 96.4% for age, education, and occupation, respectively, with random choice selection. Although using likelihood selection leads to lower diversity, it enhances the coherence score of the generated persona. This can be attributed to the fact that ChatGPT assigns low probability to illogical choices while increasing the likelihood of logical ones. We further investigate incoherent personas generated by COD and observe that the majority of cases were linked to age-occupation inconsistencies, such as an 8-year-old child working as a taxi driver. We believe this is primarily due to the current ChatGPT model's commonsense reasoning ability, rather than COD itself. It is also worth noting that likelihood selection can generate personas that better reflect real-world statistics for some attributes, compared to random selection. As shown in Figure 5, 31.7% and 27.5% of the generated personas are female and male, according to likelihood selection, while the figures for random selection are 10.6% and 8.7%.

6 Conclusion

This study addresses the issue of persona-based chatbots, specifically focusing on the automatic generation of personas. We find that ChatGPT generates personas biased towards a particular demographic group, indicating a sampling bias. To tackle this, we implemented two prompting strategies: Chain-of-decision and Listing sampling. These strategies substantially enhanced the diversity of generated personas and helped reduce biases. Importantly, our approach is versatile and does not require additional training, making it applicable to different tasks that leverage the power of LLMs.

Limitations

Chain-of-decision is a time-consuming and computationally expensive process of listing choices and estimating their likelihoods. This complexity poses challenges when applied to larger LLMs such as GPT-4. Additionally, limitations arise in the choice listing step, where the model might struggle to list all possible options due to generation length constraints. This limitation explains why the coverage rate for certain attributes, like nationality (with up to 196 possibilities), remains low at 45.9%.

In the case of Listing sampling, requesting diverse responses from the language models leads to longer generated texts and, therefore, increases latency. This latency issue is a significant concern for applications like chatbots that require a very fast response time.

Finally, while the suggested methods minimise bias and enhance diversity, it remains an open question about how the generated personas truly mirror real-world data. For example, using a Chain-of-decision with random selection approach might lead to an over-representation of uncommon genders. This highlights the need for a more refined metric that balances diversity and real-world statistics.

Ethical considerations

The development of persona-based chatbots using LLMs like ChatGPT poses two ethical risks: (1) **Stereotyping bias:** While this paper focuses on addressing the sampling bias of LLMs, it is essential to note that other forms of stereotyping bias might persist. Persona-based chatbots created through this method could display prejudice or discrimination towards specific demographic groups. Potential solutions to tackle these stereotyping biases are discussed in Section 2. (2) **Privacy concerns:** Creating individual personas using LLMs raises privacy concerns. Since LLMs are trained on extensive datasets, there is a risk that private individual information might be included in the training data. Consequently, the persona generation process could inadvertently leak sensitive information about real individuals. To mitigate this, pseudonymisation techniques (Pilán et al., 2022; Lison et al., 2021; Ovalle et al., 2023) can replace personally identifiable information, such as named entities, in the training data.

References

Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 298–306.

Jaewoo Ahn, Yeda Song, Sangdoo Yun, and Gunhee Kim. 2023. Mpchat: Towards multimodal persona-grounded conversation. *arXiv preprint arXiv:2305.17388*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, and Xinchao Xu. 2020. Plato-2: Towards building an open-domain chatbot via curriculum learning. *arXiv preprint arXiv:2006.16779*.

Yu Cao, Wei Bi, Meng Fang, Shuming Shi, and Dacheng Tao. 2022. A model-agnostic data manipulation method for persona-based dialogue generation. *arXiv preprint arXiv:2204.09867*.

CharacterAI. Character ai. https://beta.character.ai/.

John Joon Young Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. *arXiv preprint arXiv:2306.04140*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Wanyu Du, Jianqiao Zhao, Liwei Wang, and Yangfeng Ji. 2022. Diverse text generation via variational encoder-decoder models with gaussian process priors. *arXiv preprint arXiv:2204.01227*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633*.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2023. Bias and fairness in large language models: A survey. *arXiv preprint arXiv:2309.00770*.

Aparna Garimella, Akhash Amarnath, Kiran Kumar, Akash Pramod Yalla, N Anandhavelu, Niyati Chhaya, and Balaji Vasan Srinivasan. 2021. He is very intelligent, she is very beautiful? on mitigating social

biases in language modelling and generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4534–4545.

Aparna Garimella, Rada Mihalcea, and Akhash Amarnath. 2022. Demographic-aware language model fine-tuning as a bias mitigation technique. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 311–319.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Seungju Han, Beomsu Kim, Jin Yong Yoo, Seokjun Seo, Sangbum Kim, Enkhbayar Erdenee, and Buru Chang. 2022. Meet your favorite character: Open-domain chatbot mimicking fictional characters with only a few utterances. *arXiv preprint arXiv:2204.10825*.

Xudong Han, Timothy Baldwin, and Trevor Cohn. 2021. Balancing out bias: Achieving fairness through balanced training. *arXiv preprint arXiv:2109.08253*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Gibbeum Lee, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, and Kangwook Lee. 2023. Prompted llms as chatbot modules for long open-domain conversation. *arXiv preprint arXiv:2305.04533*.

Pierre Lison, Ildikó Pilán, David Sánchez, Montserrat Batet, and Lilja Øvrelid. 2021. Anonymisation models for text data: State of the art, challenges and future directions. 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 4188–4203.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. *arXiv preprint arXiv:2010.12884*.

Justus Mattern, Zhijing Jin, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schölkopf. 2022. Understanding stereotypes in language models: Towards robust measurement and zero-shot debiasing. *arXiv preprint arXiv:2212.10678*.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. *arXiv preprint arXiv:1809.01984*.

OpenAI. 2023. Gpt-4 technical report.

Anaelia Ovalle, Kai-Wei Chang, Ninareh Mehrabi, Yada Pruksachatkun, Aram Galystan, Jwala Dhamala, Apurv Verma, Trista Cao, Anoop Kumar, and Rahul Gupta. 2023. Proceedings of the 3rd workshop on trustworthy natural language processing (trustnlp 2023). In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*.

Vishakh Padmakumar and He He. 2023. Does writing with language models reduce content diversity? *arXiv preprint arXiv:2309.05196*.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. 2022. The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101.

Rebecca Qian, Candace Ross, Jude Fernandes, Eric Smith, Douwe Kiela, and Adina Williams. 2022. Perturbation augmentation for fairer nlp. *arXiv preprint arXiv:2205.12586*.

Replika. Replika ai. https://replika.com/.

Stephen Roller, Y-Lan Boureau, Jason Weston, Antoine Bordes, Emily Dinan, Angela Fan, David Gunning, Da Ju, Margaret Li, Spencer Poff, et al. 2020. Open-domain conversational agents: Current progress, open problems, and future directions. *arXiv preprint arXiv:2006.12442*.

Danielle Saunders, Rosie Sallis, and Bill Byrne. 2021. First the worst: Finding better gender translations during beam search. *arXiv preprint arXiv:2104.07429*.

Patrick Schramowski, Cigdem Turan, Nico Andersen, Constantin A Rothkopf, and Kristian Kersting. 2022. Large pre-trained language models contain human-like biases of what is right and wrong to do. *Nature Machine Intelligence*, 4(3):258–268.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. *arXiv preprint arXiv:1908.06605*.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2020. Towards controllable biases in language generation. *arXiv preprint arXiv:2005.00268*.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. Blenderbot 3: a deployed conversational agent that

continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.

Irene Solaiman and Christy Dennison. 2021. Process for adapting language models to society (palms) with values-targeted datasets. *Advances in Neural Information Processing Systems*, 34:5861–5873.

Yixuan Su and Nigel Collier. 2022. Contrastive search is what you need for neural text generation. *arXiv preprint arXiv:2210.14140*.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Pranav Narayanan Venkit, Sanjana Gautam, Ruchi Panchanadikar, Shomir Wilson, et al. 2023. Nationality bias in text generation. *arXiv preprint arXiv:2302.02463*.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*.

Tae-Jin Woo, Woo-Jeoung Nam, Yeong-Joon Ju, and Seong-Whan Lee. 2023. Compensatory debiasing for gender imbalances in language models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

Ran Zmigrod, Sabrina J Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. *arXiv preprint arXiv:1906.04571*.

A Appendix

A.1 Prompt templates

A.1.1 Chain-of-decision

Prompt template for decision listing:
I want to create a random individual profile that includes demographic information. What are the key decisions I need to make?
Please keep the number of decisions minimal. Kindly structure your reply in the following format: ` <#number>. <decision name>: <decision description>`

Prompt template for choice listing without likelihood estimation:
I want to create a random individual profile that includes demographic information.
The following information has been determined:
<Decision name #1>: <Selected choice #1>
...
<Decision name #i>: <Selected choice #i>
Conditioning on the above information, please list a diverse range of possible choices for <Decision name #(i+1)> (as many as possible). Kindly structure your reply in the following format: ` <#number>. <choice name>`

Prompt template for choice listing with likelihood estimation:
I want to create a random individual profile that includes demographic information.
The following information has been determined:
<Decision name #1>: <Selected choice #1>
...
<Decision name #i>: <Selected choice #i>
Conditioning on the above information, please list a diverse range of possible choices for <Decision name #(i+1)> (as many as possible), each accompanied by your approximate of likelihood, rated on a scale from 0% (highly improbable) to 100% (highly probable). Ensure the total probabilities sum up to 100%. Kindly structure your reply in the following format: ` <#number>. <choice name> - <choice probability>`

Prompt template for content generation:
I want to create a random individual profile that includes demographic information.
The following information has been determined:
<Decision name #1>: <Selected choice #1>
...

<Decision name #N>: <Selected choice #N>

A.1.2 Listing sampling

Prompt template for persona revealer (i.e. chatbot), who will reveal his/her persona during the conversation:
Given this conversation:
...
Person B:
Person A:
Imagine you are person B and act as if you were a real individual who willing to disclose everything. Consider various ways in which person B might response to continue the conversation, and then provide a list of #N different diverse responses. Please keep these responses short.

Prompt template for persona seeker, who will extract persona information from the other speaker:
Given this conversation:
...
Person A:
Person B:
Imagine you are person A and act as if you were a real individual. Your goal is to guide the conversation towards extracting basic demographic information that includes age, gender, nationality, occupation, and level of education from Person B. Ensure that the topic transition feels smooth. Please keep your response short with no more than two sentences.

A.1.3 Persona attribute extraction

Prompt template for attribute extraction from a generated profile (In-advance generation):
Given this profile:
<Profile description>
Please use the information above to complete the following details. For any missing information, please fill in 'None'.
Age:
Gender:
Nationality:
Place of birth (country):
Highest education:
Occupation:

Prompt template for attribute extraction from a conversation (On-the-fly generation):
Given this conversation:
...
Person A:
Person B:
Please extract information about person B from the conversation and complete the following details. For any missing information, please fill in 'None'.
Age:
Gender:
Nationality:
Place of birth (country):
Highest education:
Occupation:

Prompt template for mapping an extracted attribute value to a pre-defined value as defined in Appendix A.2:
<Attribute name>: <Extracted attribute value>
To which group does the above <Attribute name> belong? Give your answer without any explanation. Return "others" if it does not fit into any specific category listed.
Pre-defined value #1
...
Pre-defined value #N

A.2 Pre-defined attribute values

Table 3 shows the pre-defined values for each of the persona attributes. The values for gender are extracted from the Wikipedia page[3]. The values for the occupation sector are extracted from this web page[4]. The values for the highest education are determined by querying ChatGPT, as well as examining the original education descriptions in the generated personas.

[3] https://en.wikipedia.org/wiki/List_of_gender_identities
[4] https://www.prospects.ac.uk/jobs-and-work-experience/job-sectors

Attributes	Pre-defined values	Count
Age group	0-10, 10-20, 20-30, 30-40, 40-50, 50-60, 60-70, 70+	8
Gender	Abinary, Agender, Ambigender, Androgyne, Androgynous, Aporagender, Autigender, Bakla, Bigender, Binary, Bissu, Butch, Calabai, Calalai, Male, Female, Demigender, Demiflux, Dual gender, Femme, Genderfae, Genderfluid, Genderflux, Genderfuck, Genderless, Gender non conforming, Genderqueer, Gender questioning, Graygender, Hijra, Intergender, Intersex, Kathoey, Maverique, Meta gender, Multigender, Muxe, Neurogender, Neutrois, Non binary, Omnigender, Pangender, Polygender, Sekhet, Third gender, Transgender, Transsexual, Travesti, Trigender, Tumtum, Two spirit, Vakasalewalewa, Waria, Winkte, X gender, Xenogender, Prefer not to say	57
Nationality	All 196 nationalities	196
Highest education	No formal education, Primary school, Secondary school, High school, Associate Degree, Certificate programs, Diploma, Bachelor, Master, PhD, Doctorate Degree, Juris Doctor, Medical Doctor	13
Occupation sector	Accountancy, banking and finance Business, consulting and management Charity and voluntary work Creative arts and design Energy and utilities Engineering and manufacturing Environment and agriculture Healthcare Hospitality and events management Information technology Law Law enforcement and security Leisure, sport and tourism Marketing, advertising and PR Media and internet Property and construction Public services and administration Recruitment and HR Retail Sales Science and pharmaceuticals Social care Teacher training and education Transport and logistics Student Unemployed Retired	27

Table 3: Pre-defined values for persona attributes