

University of Nottingham
UK | CHINA | MALAYSIA

One-to-Many Problems in Interactive AI Systems

Thesis submitted to the University of Nottingham for the degree of **Doctor of Philosophy, February 2025.**

Benjamin Towle

20266746

Supervised by

Ke Zhou
Xin Chen

Signature *Benjamin Towle*

Date 4 / 12 / 2025

Abstract

This thesis focuses on the one-to-many problem and the various unexplored challenges therein. Specifically, for most machine learning tasks, there is sufficient uncertainty that there is usually more than one plausible output for a given input. Failure to explicitly account for this can lead to models that are unable to represent the full range of possible outputs.

Interactive AI systems comprise systems in which the output is not just a function of a static input, but also some interaction from a user. Concretely, this includes dialogue systems, where a user interacts with a chatbot, and also interactive image segmentation systems, where the final AI-produced segmentation is guided by a user-provided bounding box or click. In these cases, the user's intentionality provides a key source of uncertainty. For instance, if a user clicks a wheel on the image of a car, do they want the AI model to segment the entire car, all four wheels, or just that single wheel? It is precisely this user-induced uncertainty that forms the focal point of this thesis.

The thesis is split into mainly two parts. In Part I, we consider implicitly one-to-many tasks – tasks where there are multiple plausible outputs, but the final task still requires predicting a single output. In Part II, we consider explicitly one-to-many tasks – tasks in which the end goal is to produce multiple outputs.

We first deal with what to do when there is a mismatch between the output the

model wants to generate and the outputs available to it. Then we handle the related problem of ensuring the model is capable of learning the full range of outputs for a particular task. We then try to generalise our methods so that they can make use of large language models, which provide powerful performance across a range of tasks without requiring bespoke training, and also across modalities such as images, so that we can show our techniques do not require wholesale redesigning if the modality changes. Lastly, in Part I we show how our techniques can handle a user that provides new information through their behaviour over time.

In Part II, we first demonstrate a simulation-based approach with smart reply systems, which, in contrast to the ad hoc previous methods, provides a principled solution to generating a set of relevant and diverse reply suggestions. Finally, we address how to deal with model architectures that are inherently deterministic (e.g. image segmentation), and we demonstrate a recurrent neural network approach that allows for multiple predictions, without requiring any major alterations on the underlying base architecture.

Overall, this thesis presents a variety of technical contributions that solve a range of challenges pertaining to the one-to-many problem in interactive AI systems, by synergising various techniques such as simulation, knowledge distillation, and bootstrapping.

Acknowledgements

My journey towards completing a PhD in computer science has charted a strange Odyssey, replete with many twists and turns. I began as a Classicist, reading *Literae Humaniores*, at St John's College, Oxford; and, although I spent some time programming as a hobby–building chatbots in Python, app development in Java, and doing some web dev as webmaster for the university tennis team–I didn't envision at the time that a transition to a full-time career in computer science would be feasible. Particularly, although on the industry side there was still an undersupply of programmers, at the research level, it seemed that it would be difficult to transition to CS from a non-technical subject.

Following completion of my undergraduate degree, I ended up working as a corporate banker for one of the top American investment banks. Yet, while the position was well compensated, I considered the question of what I would do once I had hypothetically earned enough to retire. At this, I realised that I would probably be compelled to go back to research at a university. It was at this point that I began to investigate my options for a master's degree.

To my surprise, I found that the University of Nottingham–the first to be acknowledged in this *encomium*–allowed for non-STEM students to take their master's degree in computer science. I had seen other universities offer various masters' courses to non-STEM students, but those had always been separate courses to those offered to STEM students (e.g. *introduction to CS*, etc.).

Taking this opportunity, it became clear early on in the course that I wanted to do further research into the subject, rather than just a single year's study, and I applied to the PhD program. Here, I would like to give thanks to my supervisor Ke (Adam) Zhou, who agreed to supervise me for a PhD. At the time of application, it was only a few months into starting the master's course, so although I knew some programming, my theoretical knowledge of CS was limited. I appreciate that he decided to take a chance on me, compared to other safer options that may have been available with students from a traditional CS background.

Having started the PhD, I was also introduced to Xin Chen, my other supervisor, who showed me the range of applications deep learning could have to medical imaging. I am grateful for his support, and found the fortnightly meetings with the rest of his team extremely useful, especially when there was a logjam with my particular research.

I would also like to thank the internal and external examiners, Magan Moemeni and Joemon Jose, respectively, for their care and attention in providing invaluable feedback to help improve the thesis, following my viva.

Finally, I would like to thank my parents who have supported me during this time. It was during the COVID lockdowns that I first returned to Nottingham, and being able to stay with my parents compared to the scenario of being locked down in an unfamiliar city provided an indubitable benefit to the start of my research, and their continued support whether moral or through proofreading manuscripts for typos–of which I make many–has proved invaluable over the last few years.

Publications

- **Chapter 4:** Towle, Benjamin and Ke Zhou. "Learn What Is Possible, Then Choose What Is Best: Disentangling One-To-Many Relations In Language Through Text-based Games". In: *Findings of EMNLP* (2022).
- **Chapter 8:** Towle, Benjamin and Ke Zhou. "Model-Based Simulation for Optimising Smart Reply". In: *ACL* (2023).
- **Chapter 9:** Towle, Benjamin and Ke Zhou. "End-to-End Autoregressive Retrieval via Bootstrapping for Smart Reply Systems". In: *Findings of EMNLP* (2023).
- **Chapter 6:** Towle, Benjamin, Xin Chen and Ke Zhou. "SimSam: Zero-shot Medical Image Segmentation via Simulated Interaction". In: *ISBI* (2024).
- **Chapter 7:** Towle, Benjamin and Ke Zhou. "Enhancing AI Assisted Writing with One-Shot Implicit Negative Feedback". In: *EMNLP* (2024).
- **Chapter 10:** Towle, Benjamin, Xin Chen And Ke Zhou. "SeqSAM: Autoregressive Multiple Hypothesis Prediction for Medical Image Segmentation Using SAM". In: *ISBI* (2025).

Notation

This section addresses the approaches to notation taken throughout the thesis. As multiple data types / modalities are involved in this thesis (both text and images) we have tried to ensure consistency as much as possible. In some cases however, more fine-grained notation may be used, such as indexing the words of a sentence, while in other cases we may just use the raw sentence itself.

- Strings are represented with lowercase letters, such as x, y, z. Subscripts are added, such as x_t, to represent the t-th word of said string.
- Sets of strings are represented with uppercase letters, e.g. X, Y, Z.
- Scalar hyperparameters are represented with uppercase letters (e.g. the number of pixels in an image is N).
- Vectors are represented with lowercase bold letters, such as $\mathbf{x}, \mathbf{y}, \mathbf{z}$.
- Sets of vectors and matrices are represented with uppercase bold letters, with subscripts to differentiate them as needed, e.g. $\mathbf{H}_x$.
- Probability distributions are represented with lowercase $p(\cdot)$ or $q(\cdot)$. Subscripts are used as needed to differentiate where probability distributions are obtained from different models, e.g. $p_\Theta(\cdot)$.
- Manifolds are represented with uppercase calligraphic font, such as $\mathcal{X}$ or $\mathcal{Y}$.

Contents

Abstract **i**

Acknowledgements **iii**

Publications **v**

Notation **v**

List of Tables **xiii**

List of Figures **xiv**

Abbreviations **xvii**

Chapter 1 Introduction **1**

1.1 Problem Formulation and Objectives 2

1.2 Contributions 7

1.3 Thesis Outline 9

Chapter 2 Background **12**

2.1 Applications 12

2.2 Deep Learning 13

2.3 Loss Functions 14

2.4 Transformers 16

2.5 Large Language Models 16

2.6 Zero-shot Learning 17

2.7 Knowledge Distillation 18

2.8 Latent Variable Models 19

2.9 The One-to-Many Problem 21

2.10 Evaluation 23

I Implicitly One-to-Many **26**

Chapter 3 Mode Unavailability **27**

3.1 Introduction 27

3.2 PAROR 30

3.3 Experiment 33

3.4 Conclusion 35

Chapter 4 Mode Collapse **37**

4.1 Introduction 37

4.2 Method 39

4.3 Experiment 45

4.4 Conclusion 49

Chapter 5 One-to-Many for LLMs **50**

5.1 Introduction 50

5.2 Related Work 52

5.3 Method 53

5.4 Experiment 56

5.5 Conclusion 62

Chapter 6 Different Modalities **63**

6.1 Introduction 63

6.2 Methodology 65

6.3 Results 67

6.4 Conclusion 70

Chapter 7 Dynamically One-to-Many **72**

7.1 Introduction 72

7.2 Related Work 74

7.3 Method 75

7.4 Experiment 76

7.5 Conclusion 80

II Explicitly One-to-Many **81**

Chapter 8 Optimisation for Multiple Outputs **82**

8.1 Introduction 82

8.2 Related Work 84

8.3 Framework 85

8.4 SimSR 86

8.5 Experiments 89

8.6 Conclusion 95

Chapter 9 Efficient Optimisation for Multiple Outputs **96**

9.1 Introduction 96

9.2 Related Work 98

9.3 Methodology 99

9.4 Experimental Setup 102

9.5 Results 105

9.6 Conclusion 109

Chapter 10 Deterministic Models **111**

10.1 Introduction 111

10.2 Methodology 113

10.3 Results 116

10.4 Conclusion 118

III Conclusion **119**

Chapter 11 Conclusion and Future Work **120**

11.1 Conclusion 120

11.2 Future Work 122

Bibliography **125**

List of Tables

1.1 Different translations of the first sentence of Homer's *Iliad* . . . 2

1.2 Examples of input-output pairs for different machine learning tasks 4

3.1 Sample from the PersonaChat dataset (Zhang et al., 2018a). Colour-highlights indicate semantic overlap. Through posterior term-level matching PAROR selects a response that takes both the context and most relevant profile into account, whereas the baseline bi-encoder only trivially repeats one of the profiles. 28

3.2 Results of PAROR and BERT-PAROR and previous methods on the PersonaChat test sets. †= uses pretrained model. ‡= results obtained from our run. 34

3.3 Results of PAROR and various ablations on the PersonaChat original test set. Ablations are applied separately, rather than cumulatively (see 3.3.2 for details). 36

4.1 Rules for intent detection with number of samples by intent. Matching conditions are tested in the listed order. SpaCy is used for relevant part-of-speech tagging. 43

4.2 Statistics for the ClubFloyd (CF) and Jericho Walkthroughs (JW) datasets. 45

4.3 Recall@k performance on the Jericho Walkthroughs test set. Games are classified as either 'possible' or 'difficult', using the classification from Liu et al. (2023a). †/‡ = statistically significant on t-test with p-value < 0.01 compared to CALM-FT and Student-only ablation respectively. 47

5.1 Statistics of the datasets used. 'User' here refers to the annotator for the classification datasets and the speaker for the Friends dataset. Examples for user are taken from the Train set for each respective dataset. 56

5.2 Evaluation results on the test sets for personalised classification and generation tasks, comparing the effects of baseline approaches with and without our method UWCG. **Bold** indicates best result, underline indicates second best result. 57

5.3 Ablation study on the effect of using our method, UWCG, versus vanilla classifier guidance using personalised retrieval with the **Flan-T5-Large** model. **Bold** indicates superior result for respective number of K. 61

5.4 Analysis of mean (std.) of the cosine similarity of the example retrieved by RAG as well as the pearson correlation between cosine similarity and accuracy / ROUGE-1. 61

6.1 Mean $\pm$ standard deviation of results on the Breast Ultrasound Scan, CVC-ClinicDB and ISIC-2016 test sets, showing: (A) upper bound obtained from fine-tuning; (B) zero-shot results for our method and the baseline; (C) ablations of key components from our method. **Bold** indicates best zero-shot result. ‡ indicates statistically significant difference between SimSAM and SAM using Wilcoxon Signed Rank Test ($p < 0.01$). 68

7.1 Results on MultiWOZ and SGD test sets using ROUGE-L and R@1 metrics. **Bold** indicates best result. * indicates result is statistically significant with p-value < 0.01 compared to best baseline. 77
7.2 Results on MultiWOZ and SGD test sets under different values of α, using the T5 NIFTY Intent model. 78
7.3 Human evaluation using crowdworkers. * indicates result is statistically significant with p-value < 0.01 using binomial test. . . 78
7.4 Qualitative example from the MultiWOZ test set. Text is post-processed to remove reference numbers as the model does not have access to booking API. Text in square brackets is annotated intents from dataset. 79

8.1 Results on the Reddit and PersonaChat Test sets under different search strategies for SIMSR. 89
8.2 Statistics for the datasets. 91
8.3 Performance of SIMSR (B) compared to baseline approaches (A) and ablations (C) on the Reddit and PersonaChat Test sets. All results are statistically significant on t-test with p-value < 0.01. . 93
8.4 Examples of model outputs on the PersonaChat (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. 94
8.5 Latency of SIMSR compared to baseline approaches on the Reddit Validation set. 95

9.1 Number of samples in the Train, Validation, Test sets and Candidate pool in the three datasets for evaluation. The Candidate pool comprises the Train set with duplicate responses removed. . 104

9.2 Performance of STAR across Reddit, PersonaChat and DailyDialog Test sets on relevance (ROUGE) and diversity (Self-ROUGE) metrics. **Bold** indicates best result, underline indicates second-best. * = statistically significant versus next best result on t-test with p-value < 0.01. 106

9.3 Performance of STAR on the Reddit, PersonaChat and DailyDialog Validation sets under different model configurations. Ablations are applied separately. **Bold** indicates best result, underline indicates second-best. * = statistically significant versus next best result on t-test with p-value < 0.01. 106

9.4 Example model outputs from the DailyDialog Test set, comparing STAR (ours) with the top-performing baseline method. Numbers in **bold** indicate the ranking the reply received according to the Matching model. 109

10.1 Results on LIDC-IDRI and QUBIC Kidney test sets. (A) Performance of SeqSAM versus SAM MCL baseline. **Bold** indicates best result. † indicates the result is significantly better than the baseline on Wilcoxon Signed Rank Test ($p < 0.01$). (B) Effect of various ablations on SeqSAM. 115

List of Figures

1.1 An ontology of the challenges of the one-to-many problem, showing their relationship to one another, and in which chapters corresponding contributions to their solution can be found. 5

2.1 Segment Anything Model (SAM). The model is capable of generating up to 3 unique masks and also a confidence score per mask, obtained via training to predict the overlap between its prediction and the ground-truth mask. 18

2.2 Graphical representation of normalised surface distance. Credit: https://metrics-reloaded.dkfz.de/metric?id=normalized_surface_distance 25

3.1 Comparison between cross-encoders, bi-encoders and our proposed method in personalised dialogue. Like bi-encoders, our method separately encodes responses to keep inference fast, but performs matching at the term-level and weights persona profiles according to their posterior relevance with the response. 27

3.2 Overview of the proposed PAROR model. Colour-highlights in text (left) indicate semantic overlap. 30

4.1 Overview for PASA. A teacher model learns to predict K different actions to capture the pre-training data distribution, and the posterior probability from the fine-tuning data is distilled into a student model. 37

4.2 The architecture for PASA. The teacher model is trained to predict the next action, by being given access to posterior information z about the action. In this case, the posterior information reflects some hint about what action is being taken, such as the type of action (we explore different options here). From this, the teacher outputs a probability distribution over possible actions, called 'soft targets', as opposed to 'hard targets' where p=1 for the ground truth action and p=0 elsewhere. The student then learns to imitate these soft targets, providing a richer learning environment. 40

4.3 Cumulative frequency graph of normalised player scores from the ClubFloyd dataset. 45

5.1 Effect of different numbers of retrieved examples (K) for the GoEmotions and Unhealthy Conversations datasets under different model sizes. 60

6.1 (A) Previous methods predict annotation mask in a single step; (B) SimSAM, our method, simulates possible click locations to provide an additional prompt to the model and retrieves the most compatible mask from a pool of generated masks. Example is from Breast Ultrasound Scan test set (Al-Dhabyani et al., 2019). 64

6.2 Qualitative results from the Breast Ultrasound Scan, CVC-ClinicDB and ISIC-2016 test sets. (Col. 1) Image $\mathbf{x}$ with bounding box prompt; (Col. 2) binary ground truth mask $\mathbf{y}$; (Col. 3) baseline SAM model (Kirillov et al., 2023); (Col. 4) SimSAM; (Col. 5) Union of all $K = 50$ masks generated by SimSAM. 69

7.1 Example of how an agent may utilise either a smart reply system or an AI writing system to speed up communication with a customer. Our approach, NIFTY, uses implicit negative feedback from the rejected suggestions to improve the AI writer's prediction. 73

8.1 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. 83

9.1 Previous methods [A] compared to our approach, STAR [B]. The example displayed is taken from the DailyDialog Test set, and compares the predictions of STAR with SimSR (Towle and Zhou, 2023b), the next best method. Our method's suggestions present a diverse range of topics/intents to drive the conversation. 97

9.2 Comparison of overall relevance and diversity scores across ablations, obtained by averaging across all three datasets with equal weighting. 107

9.3 Comparison of run-time efficiency between STAR and the baseline methods. Results are calculated over the Reddit Validation set. 108

10.1 Overview of SEQSAM. A Recurrent Module sequentially generates masks. When more predictions than labels are generated, we chunk the predictions into groups and perform subsampling. . 112

10.2 Qualitative results on LIDC-IDRI test set using SAM, showing annotated lung nodules. (Col. 1) Input images; (Col. 2-5) labels from multiple human annotators; (Col. 6-8) MCL baseline (Kirillov et al., 2023); (Col. 9-11) SEQSAM, our method, for $M = 3$. 116

Chapter 1

Introduction

Machine learning, at its core, is the process of learning patterns from a set of data. This broad term encompasses a family of various types of learning, depending on the nature of the task: *supervised learning*–perhaps the most well known–involves learning to predict an output, given an input. This makes itself at home in tasks such as machine translation, such as predicting an output French sentence from an input English sentence. *Unsupervised learning*, by contrast, lacks a pre-defined output, and involves learning patterns simply from the input data alone. An example of such may be normative models used in the medical field, where novel inputs that do not match the distribution learned may be flagged as anomalous. Finally, there is reinforcement learning, which involves learning what actions an agent should take in order to maximise some reward. Like unsupervised learning, it lacks a pre-defined output, but instead this must be discovered through trial-and-error in some environment. The most well-known example of this in the public eye is perhaps AlphaGo, which in 2016 defeated multiple-time world champion Go player Lee Sedol, a task previously thought impossible with the technology of the time. These three are by no means exhaustive, but represent three of the most common paradigms a reader of machine learning literature is likely to encounter. Table 1.2 shows several further examples of various problems amenable to machine learning.

Only a couple of decades ago, machine learning was focused on largely academic and, by modern standards, simple tasks, such as classifying an image of a digit into one of ten numbers, or predicting whether an image contained a cat or a dog. Nowadays however, machine learning is called upon to solve a much wider range of complex and more high-stakes tasks. Machine learning is now used to drive cars, assist in medical diagnoses, and even in national security. As of now, large language models (LLMs) have become a household concept, testifying to how deeply culturally and economically ingrained machine learning has become. Indeed, OpenAI's ChatGPT as of 2025 has now reached the verb-like status of Google.

Original: **"Menin aeide thea, Peleiadeo Achileos" (Greek)**
Translation 1: "Sing, goddess, the anger of Peleus' son Achilles"
Translation 2: "Rage—Goddess, sing the rage of Peleus' son Achilles"
Translation 3: "Sing, O goddess, the destructive wrath of Achilles son of Peleus"
Translation 4: "The wrath sing, goddess, of Peleus' son Achilles"

Table 1.1: Different translations of the first sentence of Homer's *Iliad*

1.1 Problem Formulation and Objectives

Despite all of these advances, machine learning systems are typically unable to attain 100% accuracy across many tasks. This is not entirely due to architectural problems or a lack of data on their part however, but rather due to the structure of many of the tasks they are faced with. To explain this, recall previously that we mentioned supervised learning involves predicting an output given an input. Unfortunately however, this is an oversimplification of the underlying structure of many supervised tasks. For, there may often be many valid outputs for a given input. Take the case of translation, for instance; the best way to translate a particular passage often depends heavily on stylistic choices made by the translator. Due to the imperfect linguistic and cultural mapping between languages, there may be extensive disagreement between any two translators over how 'best' to translate a passage. Table 1.1 shows a classical example, displaying various competing translations of the first line to Homer's Iliad, a text overall whose 'correct' translations have been hotly contested, even to the point that famous lectures have been given on them (Arnold, 1861). While the focus of this thesis is not at all focused particularly on theories of translation, the fact that the one-to-many problem was one faced by humans (and still unsolved!) that long predates even the first precursors of machine learning, is a testament to the significance and challenges of the problem that machine learning will have to contend with.

In the context of machine learning, the impasse illustrated above is termed the **one-to-many problem**, and its challenges, how they arise, and the solutions to address them form the crux of this thesis. As indicated earlier, the supervised learning paradigm, which dominates much of machine learning is greatly in tension with problems of this type due to its framing the problem as one-to-one in nature (i.e. one input = one output). Later, in Chapter 2 we deep dive the mathematics and theory behind why this is fundamentally a problem. In this chapter however, it suffices to understand the intuition behind why this framework is less than ideal. To illustrate, we furnish several more modern examples.

Consider an email reply suggestion system, such as is likely familiar with tools such as Microsoft Outlook. When receiving an email, the system generates multiple (typically three) candidate replies that may be clicked in place of requiring manual typing from a user. Naturally, such a system must go beyond simply predicting the three most likely replies from the user, and suggest replies that maximise the chance that at least one of them is relevant. For instance, if the user receives the message 'Do you want to get coffee next Friday', then suggesting {`Yes!`, `Sure!`, `Absolutely!`} is likely to be far less helpful than {`Yes!`, `No, sorry I can't`, `Maybe, let me get back to you.`}. Crucially, there is information the system lacks about the user that makes it impossible to provide an exact answer (e.g. their preferences, schedule etc.).

In another example, a clinician may wish to obtain a second opinion on how to correctly annotate the contours of a tumour before performing surgery. Crucially, this mas major medical implications, as the more broadly the contours are drawn, the more aggressive the surgery is, possibly leading to greater scarring or other medical complications. Conversely, if the contours are drawn too narrowly, the tumour may not be fully removed, leading to relapse. Within the medical field, a multi-expert approach is already a popular approach to dealing with difficult cases involving tumour conference boards. However, in practice getting access to multiple experts is not always feasible, so a medical model that can simulate this process by providing a range of 'expert' opinions is invaluable.

In both of these cases, the model is faced with the task of providing a range of outputs that provide a non-exhaustive, but fair representation of the range of views on a particular input. However, as emphasised before, supervised learning models are not typically designed to optimise this distribution of views, but rather to provide the most likely view. In the two examples above, it should be apparent by now that that is insufficient.

However, often the input does not contain all the information required to predict the output with 100% accuracy. In language translation, for instance, words in one language often do not have a perfect 1-to-1 match with another. As such translators are called upon to make many design choices which cannot be directly inferred from the source material alone. Table 1.1 shows an example of this, depicting the first line from Homer's *Iliad*, which over the years has seen numerous competing translations. Often the trade-offs reflect the tension between translating the exact semantic meaning of each word versus translating the underlying artistic intent.

Overall, this is loosely referred to as 'uncertainty', and this problem forms the crux of the motivation for this thesis. The result of this uncertainty, as Table 1.1 implies, is that there are often multiple plausible outputs for a given input. This is referred to as the **one-to-many problem**, and, as the name implies, there are numerous challenges associated with it. Despite this, it is largely understudied, with only a handful of papers addressing the problem (Guzmán-Rivera et al., 2012; Rupprecht et al., 2016; Zhao et al., 2017a).

Input	Output
Image of a dog	Label: Dog
Text: “Translate this to French”	French Translation
Sound recording of a cat	Label: Cat meow
House features (size, location)	Predicted Price
Email content	Label: Spam/Not Spam

Table 1.2: Examples of input-output pairs for different machine learning tasks

In this thesis, we are concerned with interactive AI systems specifically. An interactive AI system is one in which rather than the AI operating autonomously to achieve a task, a human is present at some stage in the loop. This includes a human having a conversation with a chatbot or in image segmentation involves a human providing a bounding box around the object they want to be segmented in an image. The one-to-many problem is particularly relevant here because the forms of interaction provided by the user typically only give an imperfect insight into the user's intentionality and goals. As such, the user typically becomes a great source of uncertainty in the prediction process.

For instance, when interacting with a chatbot, there may be large amounts of information about the user that the chatbot does not have access to, such as the user's personality, their preferences or their objective in having the conversation. Instead, the only information the chatbot may have access to is the textual messages sent by the user. In another example, in interactive image segmentation, the user does not provide the segmentation directly, but instead only provides hints to the model such as through clicks or bounding boxes. However, this form of interaction is again often highly ambiguous. For example, given an image of a car, if a user clicks the wheel of the car, should the model segment just that wheel, all the wheels, or the entire car? It is precisely this gap between the user's observed behaviour and their underlying intentionality that makes interactive AI systems a particularly interesting setting to explore the one-to-many problem.

In this thesis, we plot the course through an interconnected series of challenges that comprise the one-to-many problem (Figure 1.1). At a coarse level, the challenges can be roughly partitioned into two groups depending on the nature of the task being solved. (1) Tasks in which the end goal is to produce a single output, but in which the relation between the inputs and outputs is one-to-many; (2) Tasks which directly require predicting multiple outputs.

Problems of type (1) are unique in that though the task requires production of a *single* output, there are *multiple* plausible outputs per task. Throughout this thesis, we refer to this as *implicitly one-to-many*. Inherently, this results in a challenging machine learning task, as the default training scenario assumes a single output for a given input.

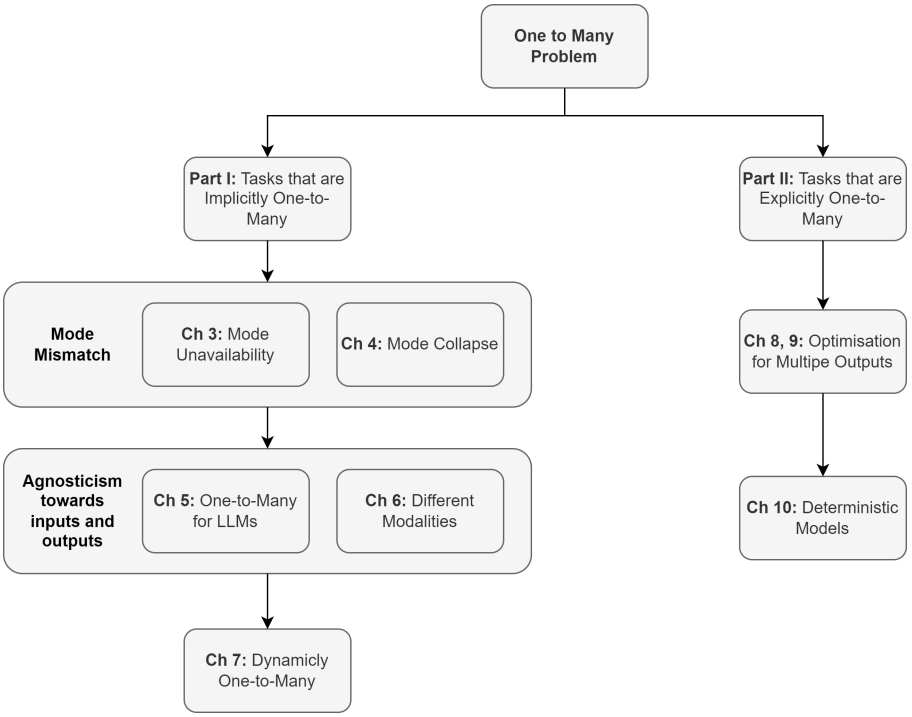


Figure 1.1: An ontology of the challenges of the one-to-many problem, showing their relationship to one another, and in which chapters corresponding contributions to their solution can be found.

1.1.1 Mode Mismatch

As we deep dive this problem (1), we initially encounter two interrelated problems. Firstly, there is what we refer to as 'mode unavailability'. This occurs when there are multiple plausible outputs for a given input. For, in some tasks, the number of valid outputs is limited. This is especially true in retrieval tasks, e.g. web search, where given a query, the model can only retrieve a web page if it exists. When a query is ambiguous, however, a model may try to retrieve a hypothetical web page that could exist, even if it does not, resulting in an overall worse search experience. We address this problem in Chapter 3.

Whereas the former problem was concerned with the unavailability of appropriate outputs, the next problem can be seen as almost the opposite problem, namely 'mode collapse'. This is the process by which a model does not learn the full range of possible outputs for an input, but instead learns only a handful. This is common in cases where there is a long tail distribution in outputs, i.e. some outputs are much more probable than others. This causes the model to fail to learn some low-probability but still valid outputs, impacting model performance in downstream use cases. We address this problem in Chapter 4.

1.1.2 Agnosticism of Inputs and Outputs

While solving / mitigating these problems goes some way towards dealing with common challenges with the one-to-many problem, it often relies on bespoke, task-specific solutions. However, nowadays the machine learning world is heavily reliant on general purpose large pre-trained models which cannot be flexibly integrated into these more bespoke techniques. This reality gives rise to two interrelated challenges: First, can we develop one-to-many techniques that can be flexibly applied to this pre-trained model paradigm? We address this problem in Chapter 5.

However, the one-to-many problem is not simply something that occurs in text, but also in images. We therefore secondly approach the problem of solving one-to-many for different modalities. Particularly, in Chapter 6, we approach medical image segmentation, which is commonly seen as a one-to-many problem due to inter-clinician disagreement over how to precisely segment an image (Monteiro et al., 2020).

1.1.3 Dynamicly One-to-Many

Having solved the four above problems, there remains one aspect to this part of the thesis unsolved. Namely, all the previous solutions assumed that the cause of the one-to-many nature of the problem was static in nature, such as user preferences or difference in clinicians. However, over time, as a user interacts with a system, we may gain further knowledge about the user that allows us to more accurately predict the next output given an input. Chapter 7 achieves precisely this, while maintaining the same flexibility as the prior two chapters.

1.1.4 Optimisation for Multiple Outputs

In Part II, we consider the more explicit variant of the one-to-many problem: namely, when the task requires multiple outputs to be predicted. Perhaps the most obvious example of this is a smart reply system, such as seen in Microsoft Outlook or Google's Gmail. In particular, given an incoming mail, a user receives **multiple** reply suggestions which can be clicked in place of typing. The key challenge here is that the goal is for *at least one* of the suggestions to be relevant to the user, but unlike other machine learning problems, we don't necessarily care about the relevance of the 2nd or 3rd most relevant suggestion.

The first chapter for this part seeks to identify how to optimise for this goal computationally, given it deviates wildly from traditional machine learning objectives. Previous work on this tasks involves trade-offs between the relevance of the output to the input, but also the diversity between the outputs. However, this is merely a heuristic, instead of the actual goal stated above. Thus in Chapter 8,

we demonstrate that the end-goal can be directly optimised through simulation, and that diversity becomes simply a byproduct of optimising this.

A corollary to this, however, is that simulation becomes computationally expensive when done in real time. We therefore perform further work to bring the latency of our model in line with prior work. Specifically, we obtain a dataset of our prior models behaviour and train a model to imitate the results of the simulation (i.e. without having to do the simulation directly in real time) making the model significantly faster. We demonstrate this solution in Chapter 9.

1.1.5 Deterministic Models

The aforementioned two chapters demonstrate a technique that can theoretically work for any *probabilistic* model. Unfortunately however, not all models produce a probability distribution over their possible outputs. Instead, some models simply deterministically produce one output. This naturally emerges as the final challenge for the one-to-many problem. We meet this challenge through the domain of medical image segmentation. Thus, we instead show how a model can be trained to sequentially generate outputs. In this way, though it remains deterministic, it is now able to produce multiple diverse and relevant outputs.

1.2 Contributions

This thesis makes 8 core technical contributions to address the aforementioned problems. In Chapter 3, we develop a 'look before you leap' approach, that addresses the mode unavailability issue by considering which modes are actually available for a particular task, before then trying to make a prediction.

In Chapter 4, we use a teacher-student-based approach to address the issue of mode collapse. The theoretical foundations for this are introduced in Chapter 2. At an abstract level however, we prevent mode collapse by first permitting the teacher to learn multiple modes, rather than forcing it to commit to one. We do this through allowing the teacher to access 'hidden' information about the task during training. The teacher with access to this hidden information is then a more powerful teacher for the student during training. This is analogous perhaps to a human teacher using a textbook as a teaching aid when teaching a student. While the student cannot avail herself of the textbook during the closed-book final exam, their overall performance is better than if the teacher had attempted to teach the student from their own memory alone.

In Chapter 5, we begin to apply our problem to large language models (LLMs), large text generation models (often conversational) pre-trained on large amounts of text data (see Section 2.5 for additional technical background). These are the *sine qua non* of text-based AI tasks at the time of writing (2025), and as such,

showing how one-to-many techniques can be applied to this critically connects the work undertaken over these last few years to the bleeding edge of modern AI. We find that the most natural home for the one-to-many problem in these models lies in their current lack of personalisation. These models are trained in user agnostic ways–whenever a user queries them, the only information they have about the user is what is stored in the conversation history. This does not scale well as the user interacts with the LLM for an increasing period of time– a year-long conversation log is highly unpractical, as the model essentially has to 'read' the entire log each time it makes its next reply. Our contribution here is to explore different ways to personalise the model and provide an analysis of the trade-offs of different approaches. In particular, we demonstrate an auxiliary model that 'looks over the shoulder' of the LLM as it is generating its reply to decide how well that reply suits that particular user and can where needed steer the LLM in the right direction.

In Chapter 6, we present a contribution that extends the one-to-many problem to modalities outside of text. In particular, we show how awareness of the one-to-many nature of a task can be used to improve a model's accuracy. We develop a simulation-based approach that imitates the behaviour of a human clinician annotating a medical image. In so doing, we can generate multiple possible annotations for a particular image, which reflects the underlying uncertainty about how it should be annotated. We then show that combining these together to form a consensus annotation produces an overall better annotation than if we had skipped the simulation step.

In Chapter 7, we address the problem of staticness in one-to-many systems, namely the assumption that the one-to-many problem remains unchanged throughout the lifecycle of a task. We use an auxiliary model, similar to Chapter 5, except whereas that approach required a defined set of users to train on, this approach is more flexible, allowing an LLM to change its behaviour at runtime in response to new information about the user, without requiring any retraining, and without making any assumptions about the source of this new information about the user.

In Chapter 8 and 9, we present two contributions that address the problem of how to optimise a system to produce multiple outputs, when the training dataset (as is common in machine learning), only contains one output per input. In Chapter 8, we contribute a simulation-based approach that simulates possible user goals at runtime, and find the set of outputs that are most compatible on average with those goals. This enables producing a set of outputs that are both highly relevant to the user, but also cover a diverse range of possible goals. Recall earlier that uncertainty of user intentionality was often a major driver of one-to-manyness. As such, obtaining this diverse coverage is highly desirable for taming this uncertainty. In Chapter 9, we made further contributions by improving the model. In particular, we noticed that while our first iteration of the model performed increasingly well as the size of the simulation increased, as this had to be carried out at runtime, it could become too slow to run at a certain point. Therefore, we instead ran the simulation on some data offline (i.e. not during runtime) and then

trained a separate model to imitate the *results* of the simulation, without having to directly perform the simulation itself. This enabled us to run much larger simulations than would be possible at runtime, and also resulted in a model that was extremely efficient.

Finally, in Chapter 10, we addressed the problem of deterministic models. The key challenge to overcome here, was that we couldn't simply sample multiple predictions from the model in the way that we did for other chapters such as 8 and 6. Instead, we recursively fed the outputs of the model back into itself during training, each time treating the output as one of the model's predictions. This enabled us to train the model to generate an arbitrary number of predictions without any sampling required.

In summary, the contributions of this thesis are as follows: In Chapter 3, we develop a posterior-aware architecture to mitigate the problem of mode unavailability, preventing the model from trying to predict non-existent options, thus improving its predictive accuracy. In Chapter 4, we develop a knowledge-distillation approach to mitigate the problem of mode collapse, allowing the model to learn the underlying one-to-many distribution of the data. In Chapter 5, we develop a novel architecture that enables our techniques to be combined with powerful large language models, greatly increasing the flexibility of our approach. In Chapter 6, we further extended the flexibility of our approach across modalities through a user-simulation based approach that leverages the model's pixel-level uncertainty to predict user interaction behaviour. In Chapter 7, we develop an architecture that enables dynamic learning from user behaviour on-the-fly and which has the flexibility of enabling black box interactive systems to learn from one another. In Chapter 8, we develop a novel smart reply system that employs a combination of user simulation and search to discover the optimal set of replies that is most likely to contain the user's desired response. In Chapter 9, we further improve this algorithm through bootstrapping combined with a new autoregressive retrieval architecture, improving the latency for the model and also its overall performance. Finally, in Chapter 10, we design a novel architecture that enables deterministic models to produce an arbitrary number of multiple outputs which are both diverse and relevant.

1.3 Thesis Outline

Here we provide an outline of the remaining chapters of this thesis. For the six chapters whose research led to publication, we also provide a GitHub repo link to facilitate reproducibility of the results.

Chapter 2. This chapter provides relevant theoretical background that is relied

upon by later chapters. This includes popular applications of the one-to-many problem, how AI is optimised via deep learning, common training paradigms, as well as how these systems can ultimately be evaluated.

Chapter 3. This chapter introduces a solution to the first half of the mode mismatch problem prevalent in the one-to-many problem. We present a novel 'look-before-you-leap' approach.

Chapter 4. In the previous chapter we were concerned with cases where during inference the predicted answer is not available. In this chapter we are concerned with the opposite side of this coin, where the training data contains many modes that we do not want to predict during inference. We achieve this through a teacher-student framework approach. https://github.com/BenjaminTowle/PASA.

Chapter 5. While the previous two chapters had considered more academic tasks of persona-grounded dialogue and text-based games, in this chapter we focus on the very modern setting of personalised LLMs. Importantly, we address the issue of applying one-to-many techniques to much more task-agnostic settings, in contrast to the more structured tasks of the previous two chapters.

Chapter 6. As the previous three chapters have been limited to the domain of text, we show how these techniques can be generalised to other modalities. We apply this to medical image segmentation, which is often categorised as a one-to-many problem due to annotator disagreement for a given image. https://github.com/BenjaminTowle/SimSAM.

Chapter 7. This chapter presents a more generalisable approach that enables models to be personalised dynamically in an unconstrained way. We guide the model in a way that is much more modality agnostic and therefore makes fewer underlying assumptions. https://github.com/BenjaminTowle/NIFTY.

Chapter 6. Represents the beginning of Part II of the thesis and shows how to obtain a model capable of generating multiple outputs, even when the training dataset only contains a single output, using a simulation-based approach, in the context of smart reply. https://github.com/BenjaminTowle/SimSAM.

Chapter 9.3.2. This chapter approaches the same task as the previous chapter, but expands upon the simulation-based approach by running it in an offline setting to make it much more efficient and scalable. https://github.com/BenjaminTowle/STAR.

Chapter 10. This chapter finally addresses how to take models whose architecture does not permit the production of multiple outputs at all, and how they can be effectively adapted to make this possible. https://github.com/BenjaminTowle/SeqSAM.

Chapter 11. Finally, this chapter provides the conclusion, summarising all of the major contributions of this thesis.

Chapter 2

Background

In this chapter we provide the necessary theoretical and background knowledge for understanding the remainder of the thesis. We first motivate how the one-to-many problem is not simply a theoretical concept confined to theory and toy applications, but actually one that is deeply intertwined with many real-world machine learning tasks.

2.1 Applications

The one-to-many problem in the context of interactive AI systems has numerous applications in real-world systems today. In this thesis, the chapters focused on downstream applications cover primarily two of these: smart reply systems and medical image segmentation. We select these as their difference in modalities–text and image, respectively–demonstrate both the generalisability of our methods proposed in this thesis, but also how broad a range of problem types are affected by the one-to-many problem.

Smart Reply systems are likely to be familiar to many readers. They are a broad term that refers to the suggested replies in many messaging and email clients that may be clicked in place of requiring manual typing. They serve as a potential time-saving measure, in a world where messaging volume is increasing, while a greater proportion of communication has shifted to away from computer keyboards towards mobile devices whose form factor makes long-form typing less than ideal.

Naturally, the fact that these systems present a user with multiple replies makes them an obvious candidate for analysis under the lens of the one-to-many problem. The reason for presenting multiple replies as opposed to just one, lies in the fact that there is usually too much uncertainty about what the user wants to say to confidently make a single prediction. This is particularly true given smart reply is often used in cases where the user would only need to make a short re-

sponse. Consider for instance, if the user receives the message: `do you want to go for coffee this evening?`. Unless the system is endowed with a large amount user-specific information such as their schedule, their personal preferences, etc., it is unlikely to be able to make an exact judgment. However, by producing multiple suggestions that capture a broad range of intents such as {`Yes, I do`; `No sorry I can't`; `Can you do another time?`} the system is able to achieve a much higher likelihood that *at least one of the replies is relevant*. Note the final part there–we do not care that every single suggested reply is close to the user's actual intent, only that one of them is. This is what makes smart reply a particularly interesting example of a one-to-many system. For, this framework is very unusual in the context of how we typically train machine learning systems, which is to maximise top-1 accuracy.

In contrast to smart reply systems, another application of one-to-many systems is that of medical image segmentation. This task refers broadly to the goal of annotating some region of interest from a medical image (usually from some type of scan, CT scan, MRI scan, X-ray etc.). Crucially, these types of scans often produce images with much weaker contrast than natural images. As such, given the task of annotating, for instance, a tumour, there is often significant disagreement between clinicians over the exact contours of the tumour–i.e. the boundary between healthy and unhealthy tissue is often blurry and unclear. This has significant consequences for downstream treatment. For, if an excessively large area around the tumour is excavated, this becomes a more severe treatment and risks greater scarring. By contrast, an overly conservative excavation may leave behind parts of the tumour, rendering the treatment ineffective. The practice of having multiple clinicians opine on a given case is commonplace in the medical field, reflected through tumour conferences. However, having multiple clinicians opine on day-to-day cases is expensive. Therefore, an AI model capable of simulating multiple clinicians through producing multiple annotations could serve as an invaluable aid to presenting a clinician with alternative segmentation hypotheses that they may have missed. However, unlike the smart reply case, where we only cared that at least one of the suggestions was relevant, these systems should also take care that all of the suggestions are at least biologically plausible. Otherwise, the system risks misleading the clinician who is reviewing it. Later, when we discuss different evaluation methods, we show how different methods can be used to evaluate and capture the different aspects of concern for different systems.

2.2 Deep Learning

Machine learning is broadly defined as 'any computer program that improves its performance at some task T, measured by P, through experiences E' (Gao et al., 2018). Machine learning contrasts from earlier AI efforts–Good Old Fashioned AI ('GOFAI')–in that it is an empirically driven technique that learns based on the learned data, rather than through a handcrafted set of rules from a domain

expert. A particular subset of machine learning, which is used in most modern generative AI models, is known as deep learning. Deep learning typically[1] involves learning a mathematically differentiable function that maps some set of inputs $\mathcal{X}$ to a set of outputs $\mathcal{Y}$ through a function f and a vector of weights $\Theta \in \mathbb{R}^D$. This is optimised through a loss function $\mathcal{L}$ that defines the compatibility between the predicted output $\hat{y}$ and the ground truth output y. Our goal is to learn the set of theta that minimises the loss:

$$\mathbb{E}_{y \sim p(y|x), x \sim p(x)}[\mathcal{L}(f(x, \Theta), y)] \tag{2.1}$$

So long as $f(\cdot)$ is differentiable, we can improve the loss by adjusting a given weight $\Theta^{(i)}$ by obtaining its partial derivative with respect to the loss. Hence:

$$\Theta_{t+1}^{(i)} \leftarrow \Theta_t^{(i)} - \alpha \frac{\delta \mathcal{L}(f(x, \Theta), y)}{\delta \Theta_t^{(i)}} \tag{2.2}$$

Where α is the learning rate that determines the 'step' size of the update. In general, smaller learning rates lead to slower convergence, while larger learning rates risk divergent behaviour. Typically, f is not a single mathematical operation (e.g. a matrix multiplication), but rather can be represented as a directed acyclic graph (DAG) of mathematical operations stacked on top of one another. We therefore obtain the gradient of θ by repeated applications of the chain rule:

$$\frac{\delta y}{\delta x} \leftarrow \frac{\delta y}{\delta t} \cdot \frac{\delta t}{\delta x} \tag{2.3}$$

This optimisation process is conducted through a dataset containing many $\langle x, y \rangle$ pairs. Each iteration of training, we sample a batch of datapoints $\langle x, y \rangle$.

2.3 Loss Functions

While all of deep learning involves some loss function $\mathcal{L}(\cdot)$, the exact choice of the loss function depends on the type of task. In this thesis, we are entirely working within discrete spaces, and so the key loss function to be aware of cross-entropy loss:

$$\mathcal{L}_{CE} = -\log p(x) \tag{2.4}$$

In other words, we want to minimise the negative log-probability of the thing we are trying to predict. Note that by convention, we typically frame everything as a minimisation problem, rather than a maximisation problem, which aligns semantically with most of the terms one will encounter in any machine learning literature such as loss, cost, error, gradient descent, etc.. Hence, even though here we want to *maximise* the probability, we simply reframe it as minimising the negative probability. Typically, in the context of this thesis we are more likely

[1] On occasion, a part of the function may be non-differentiable, such as in reinforcement learning, however the techniques that do not feature in this thesis and so are omitted here.

to be concerned with conditional probabilities, i.e.:

$$\mathcal{L}_{CE} = -\log p(y|x) \tag{2.5}$$

where x is often something like the dialogue context, and y is the reply. The model will output some probability distribution over possible replies and will then compare this to the ground truth, which can be represented as a one-hot encoded vector with zeros in all of the invalid replies and 1 for the correct one.

One limitation of this loss function in deep learning generally is that many problems have a long tail distribution. To give a simple example, suppose the goal was to predict the beginning word of a piece of text. Most of the probability mass is tied to a handful of highly probable words, whereas most words that are theoretically possible would have close to 0 probability mass. As such, models can sometimes focus all of their efforts on these highly probably words, which gives the best reduction in loss, at the neglect of other words. While a direct solution to this is out of scope of this thesis, it is useful for the reader to be conceptually aware of this.

The task of image segmentation, which we feature in chapters 6 and 10 is also a discrete problem, as we are trying to classify whether or not a given pixel is part of the object of interest. In this sense, one could also apply cross-entropy loss (typically called binary cross-entropy loss, because there are only two classes), on a pixel-wise basis. However, this is known to be highly sensitive to imbalances between the background and foreground (the object being annotated). As such, if the object only covers a small part of the image, the model may trivially optimise the loss function by assigning a high background probability to every pixel (similar to the aforementioned long tail distribution issue). To mitigate this, a popular approach, which we also employ here, is DICE loss, which provides a more balanced loss function (Maier-Hein et al., 2022). We begin by defining the DICE score:

$$DICE = \frac{2 \times |\hat{\mathbf{y}} \cap \mathbf{y}|}{|\hat{\mathbf{y}}| + |\mathbf{y}|} \tag{2.6}$$

where $|\hat{\mathbf{y}} \cap \mathbf{y}|$ is the intersection between the predicted and ground-truth mask. $|\hat{\mathbf{y}}|$ and $|\mathbf{y}|$ are the number of pixels in the prediction and ground-truth respectively. Note that this hard DICE score can also be used for evaluating a model's performance (Section 2.10). To turn this into a loss function however, two things must happen. First, we must turn this into a minimisation problem, i.e.:

$$\mathcal{L}_{DICE} = 1 - DICE \tag{2.7}$$

Second, during training, the segmentation model does not simply output a binary mask of 1s and 0s, but instead a pixel-wise binary probability. We therefore can

use a soft approximation of DICE as follows:

$$\mathcal{L}_{DICE} = 1 - \frac{2 \times \sum_n^N p(y_n) y_n}{\sum_n^N p(y_n) + \sum_n^N y_n + \epsilon} \tag{2.8}$$

where $p(y_n)$ is the model's predicted probability for the n-th pixel, y_n is the ground-truth value for the n-th pixel, and ϵ is a small value to prevent div 0 errors.

2.4 Transformers

Since around 2019 the transformer architecture (Vaswani et al., 2017a) has become the *de facto* method for training large-scale generative models (Radford et al., 2019a; Brown et al., 2020). In contrast to earlier generative models such as RNNs (Sutskever et al., 2014), which required inputs to be processed sequentially, transformers process inputs in parallel. This allowed significant hardware synergies with GPUs, which are optimised specifically for matrix multiplication.

While several variations of the transformer architecture exist, they are almost all characterised by an attention matrix:

$$\text{Attention}(Q, K, V) \leftarrow \text{softmax}(\frac{QK^T}{\sqrt{d_k}}V) \tag{2.9}$$

Where Q are the queries, K are the keys, V are the values and d_k is the dimensionality of the key. Abstractly, each query compares its similarity to each key, and selects that much of the value in proportion to its similarity. d_k serves as a regularisation parameter, as the magnitude of the dot product between two vectors is likely to be higher the larger the vectors' dimensionality. Two most popular variants of this attention are self-attention and cross-attention. Self-attention is when Q, K, V are all derived from the same inputs. Whereas, in cross-attention Q is derived from one set of inputs and K, V from another. Some types of transformer such as the encoder-decoder, which features extensively in this thesis, uses both types.

2.5 Large Language Models

Large language models use the transformer as their underlying architecture and are often trained on billions of words of large-scale data (typically from the web). Size-wise, they are often in the order of billions of parameters, frequently being too large to run on traditional consumer GPU architecture. At their core, they are optimised with a simple objective function. Given a sequence of words, predict

the next word:

$$p(x_t|x_{<t}) \tag{2.10}$$

This training represents stage 1 of their training regime. Stage 2 involves what is known as reinforcement-learning from human feedback (RLHF). As the exact details of this are somewhat out-of-scope for this thesis, we only provide a high-level explanation. Within the thesis, only Chapter 5 uses models trained via RLHF, and the choice to use them does not form a core part of the contribution of the chapter.

Essentially, given a query, e.g., `how do you bake a cake?`, the LLM will generate multiple possible answers to this. A human annotator will then provide some scoring to these answers about how well they are aligned to that humans preferences. E.g. easy-to-follow, clear instructions may receive a higher score than opaque ones. We can then upweight the LLM's likelihood for the good answer and downweight the bad answer, using a process known as reinforcement learning.

Ultimately, rather than a model that simply predicts the most likely next word from the original internet data it was trained on, we end up with a more useful model that provides helpful, human-aligned replies to the person it is interacting with.

2.6 Zero-shot Learning

One of the attractive properties of LLMs is their ability to perform what is known as in-context learning (Agarwal et al., 2024). Specifically, instead of training the model on a new task, simply by providing one or more textual examples within the models context window, it is often able to obtain powerful generalisation capabilities. The limit of this, where we do not provide the model with any textual examples, is called zero-shot learning. For instance, a sufficiently generalised language model may be able to accurately respond to `translate English to French: I have a cat`, within being explicitly trained on a dataset of English, French language pairs. These types of models have been demonstrated across a range of domains, and inevitably involves some large-scale pre-training.

One particularly relevant example for our case is Segment Anything Model (Kirillov et al., 2023), a pre-trained segmentation model released by Meta, which we use in Chapters 6 and 10. The model is trained on 1B segmentation masks from 11M images, likely the largest segmentation dataset at the time. Interestingly for this thesis, it is an interactive segmentation system, and hence many of the uncertainties around user intentionality are also present. The model works by first encoding an image into some latent representation. Then, the user can provide a variety of modalities of interaction in order to guide the model on which part of the image should be segmented, such as providing clicks, bounding boxes, or text (the 'prompt'). Because most of the model's parameters are tied up in the

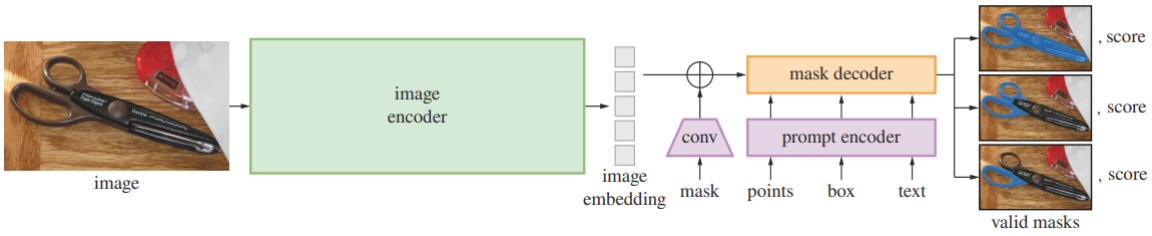


Figure 2.1: Segment Anything Model (SAM). The model is capable of generating up to 3 unique masks and also a confidence score per mask, obtained via training to predict the overlap between its prediction and the ground-truth mask.

image encoder, with a lightweight decoder, once the image has been loaded into the model, it is possible to rapidly make multiple different segmentations. This enables multiple different objects to be segmented, simply by changing what prompt is provided. Figure 2.1, taken from the original paper (Kirillov et al., 2023), shows the overall pipeline. The model has seen particular interest in the medical imaging community (Ma et al., 2023; de Oliveira et al., 2023; Roy et al., 2023) because of its zero-shot capabilities. Datasets in medical imaging are typically much smaller, because of privacy constraints, as well as the fact that it is much more expensive to get a trained clinician to annotate, say, a tumour, versus getting a crowdworker to annotate a car in a natural image. Therefore, SAM offers the enticing opportunity to be able to bridge that performance gap by providing a strong foundation that sidesteps the data size limitations of medical images. Note however, that as SAM is only trained on natural images, there is a notable performance drop when switching onto medical images, given medical images often have very different colour and contrast from natural images. This is partly what has motivated the research into SAM, as there is currently no exact consensus on the best way to apply SAM to medical images.

2.7 Knowledge Distillation

Knowledge distillation was originally introduced by Hinton et al. (2015a). At a high-level, this involves transferring the knowledge of a teacher model Φ into a student model Θ. Concretely, this involves training the student model to imitate the behaviour of the teacher model. This commonly involves training the student model to align its output probabilities, with the soft probabilities provided by the teacher:

$$\mathcal{L}_{KD} = p_{\Phi}(x) \cdot \log p_{\Theta}(x) \tag{2.11}$$

The intuition is that this allows the student model to learn additional knowledge contained in the teacher model, as the student is told the likelihood of every possible class, rather than just seeing which class was sampled, as would be the case in standard supervised learning. This is sometimes referred to as 'dark knowledge' (Hinton et al., 2015b), as it represents knowledge that the model has that

is not immediately visible to a human inspecting its behaviour. Numerous works have found this allows student models to achieve similar accuracy to their teachers but with far fewer parameters (Sanh et al., 2019a; Jiao et al., 2020; Li et al., 2021). Knowledge distillation as a technique is particularly interesting from a one-to-many perspective, as it potentially allows the student model to explicitly learn the multiple possible outputs via the teacher's probabilities, rather than having to learn from implicitly from one-hot labels.

In practice, two additional augmentations are usually made to Equation 2.11. First, the teacher probabilities are re-normalised by a temperature parameter τ, which is usually set to a high value. This is because absent this the teacher probabilities often output close to a one-hot probability distribution (i.e. 1.0 for the teacher's predicted class, 0.0 for everything else). Second, the model is also trained on the ground-truth labels–this ensures the model does not deviate far from the ground-truth in cases where the teacher may be inaccurate. This results in the following final loss function:

$$\mathcal{L}_{KD} = \text{softmax}(\eta_\Phi, \tau) \cdot \log p_\Theta(x) + \log p_\Theta(x) \tag{2.12}$$

where η_Φ are what we call the 'logits', the unnormalised scores before being transformed into a probability, and τ is the temperature parameter that dictates how flat or pointy the resultant probability distribution is. At the limit, as $\tau = 0$ the probability distribution will always assign 1 to the highest logit and 0 to everything else. As $\tau = inf$, the probability distribution will assign equal probabilities to all logits. This parameter is typically introduced as sometimes the teacher can assign an overly sharp probability distribution, that prevents the student from properly learning from the full range of the distribution (Hinton et al., 2015b). Various other extensions to this equation exist that involve matching other aspects of the two models to provide an even deeper reward signal, such as their attention matrices (Jiao et al., 2020), but those are outside of the scope of this thesis.

Adjacent to this 'soft' knowledge distillation, is a form of 'hard' knowledge distillation known as bootstrapping. In particular, bootstrapping involves training a student model on the sampled outputs of a teacher model. Previous works show how this can be used to generate synthetic datasets about a specific task, which a smaller specialist model can be fine-tuned on (Schick and Schütze, 2021b). In cases where we only have access to the output text from a model, such as if it is behind an API, then soft knowledge distillation may not be possible, so boostrapping becomes a viable alternative.

2.8 Latent Variable Models

Latent variable models have been proposed by several previous works as an approach for handling the one-to-many problem (Zhao et al., 2017b, 2018, 2019a).

In brief, instead of directly parameterising a variable X, latent variable models assume that X is conditioned on some additional unobserved variable Z. Hence:

$$p(x) = \int p(x|z)p(z) \tag{2.13}$$

In the case of a conditional generative model, which we are usually concerned with, this then becomes:

$$p(y|x) = \int p(y|x,z)p(z|x) \tag{2.14}$$

The latent variable can be treated as either a continuous or a discrete latent variable. In this thesis, we focus on the discrete case. As we show in subsequent chapters, this allows for a much richer range of options for what this underlying latent variable could represent. Besides, as we are dealing with a discrete latent variable, our integral can be tractably reduced to a probability-weighted summation over possible values of Z:

$$p(y|x) = \sum_{k}^{K} p(y|x,z_k)p(z_k|x) \tag{2.15}$$

2.8.1 Conditional Variational Autoencoder

A popular architecture for implementing latent variables is a conditional variational autoencoder. In particular, this comprises a recognition network, alongside a prediction network. During training, the recognition network has posterior access to y and x, which is used to predict $p(z|x,y)$. The prediction network then attempts to predict $p(z|x)$ and is trained using Kullbach-Leibler divergence (KLD):

$$\mathcal{L}_{KLD} = \sum_{k}^{K} p(z_k|x) \log \frac{p(z_k|x)}{p(z_k|x,y)} \tag{2.16}$$

Note the correspondence between this and the equation for knowledge distillation. In both cases, the probabilities from the teacher (recognition) network are distilled into the student (prediction) network, however the positions of the probabilities have 'swapped' places between the two equations. The knowledge distillation case can is referred to as forward KLD and the CVAE case is reverse KLD. In practice, the former leads to mean-seeking behaviour, while the latter is mode-seeking. By mean-seeking, we mean that, given a distribution with multiple peaks, the model will simply try to find the average point, even if that point sits between two peaks. By mode-seeking however, the model will try to find the largest peak. This is directly analogous to the statistical mean and mode. In other words, if the posterior network identifies multiple modes of output, the recognition network should learn to generate from only one of those modes, rather than trying to average over possible modes. In the context of a downstream task such as dialogue, the aim is that this will lead to the model producing fewer generic

responses, as generic responses are assumed to result from the model selecting words that are common across multiple modes (Zhao et al., 2017b). Note that this mean-seeking behaviour applies also to the default supervised learning case. See Pang and He (2020) for an attempt to make this training mode-seeking instead.

In addition to KLD, the recognition network is also trained to align its predictions with the ground-truth. Note that in the discrete case, although sampling would normally be non-differentiable, we can mitigate this through the gumbel-softmax trick, which allows for differentiable sampling. During inference, we discard the recognition network, and sample from the prediction network.

2.9 The One-to-Many Problem

When dealing with the one-to-many problem, our thesis has two main areas of focus. (i) explicitly representing this one-to-many-*ness* in order to improve model performance in tasks that involve predicting a single output; (ii) designing architectures to explicitly predict multiple outputs, where that is the end task. To this end, we similarly structure this section to address related work for each of those two areas.

2.9.1 Improving Model Performance

Early work on explicitly representing the one-to-many problem involved continuous latent variable models (Zhao et al., 2017a). In the case of dialogue generation, this means that rather than learning what the next reply is likely to be, it learns a distribution over possible replies. While theoretically appealing, the competing objectives in the twin loss function often resulted in posterior collapse, a known problem in CVAEs in which the model collapses to predicting the same mean and standard deviation regardless of the input. To circumvent this, further work from Zhao et al. (2018) inherently information-destroying nature of the standard CVAE loss function, and introduce a new normalisation term to the loss function. In brief, rather than constraining each sample to being close to the prior, they only require the average over the batch to be close to the prior. This intuitive approach overall leads to a much more expressive model. They also replace the continuous variable with a discrete one, which overall seemed to provide more stability. More recent approaches (Bao et al., 2020) do away with the CVAE loss function entirely, and only train a posterior network. Then, they use the model's own next reply prediction network, which it is multitask trained on, in order to select the best response. This has the advantage of removing the information-destroying term, however it requires K replies to be generated, where K is the size of the latent variable, and so scales poorly for large values of K.

Another line of work more tightly related to our thesis grounds the latent variable in some interpretable property. E.g. In the PERSONACHAT dialogue dataset (Zhang et al., 2018a), each speaker is assigned a persona comprising a few short sentences. Each sentence may be interpreted as a separate value that our discrete variable can take (Shuster et al., 2021a). While persona is a popular source of grounding in open-domain dialogue, in task-oriented and negotiation dialogue, it is more popular to use dialogue intent to reflect the variability in viable responses (He et al., 2018).

2.9.2 Generating Multiple Outputs

The task of generating multiple outputs, often referred to as Multiple Choice Learning (MCL) or Multiple Hypothesis Prediction (MHP), is not simply about sampling multiple predictions from a given model. Instead, it is closer to a set prediction problem as the predicted outputs are interdependent on each other. This is demonstrated by the objective function for MCL tasks which seeks to minimise $\mathbb{E}_{y\sim\mathbf{p}(y|x)} \max_{\hat{y}\in\hat{\mathbf{y}}} d(y, \hat{y})$ where $d(\cdot|\cdot)$ is an arbitrary distance function, and $\hat{y} \in \hat{\mathbf{y}}$ is the set of model predictions. In deep set prediction, once could generate multiple predictions and then perform a matching process in which each prediction is matched to its nearest label (Zhang et al., 2019a). Typically however, we lack multiple labels for each input. In this sense, the problem becomes closer to a semi-supervised problem.

Historically, MCL would generate multiple outputs in parallel and then only backpropagate through the most accurate head (Guzmán-Rivera et al., 2012). Unfortunately, sparse backpropagation techniques do not provide the same theoretical guarantees of convergence as dense ones. For, one prediction head may quickly become more accurate than the others, causing the other prediction heads to never receive a gradient. Some works have attempted to address this, such as Firman et al. (2018), who introduce a catch-up loss whereby the worst performing head also receives a gradient. However, this results in the trade-off that the model is no longer optimising for what will be its end goal during inference. A further limitation of standard MCL is that the number of outputs has to be fixed during training. We found this to be a major limitation in cases where one wished to finetune additional heads onto a pre-trained MCL model. For instance, the Segment Anything Model, a transformer-based image segmentation model, uses MCL to output three segmentation masks (Kirillov et al., 2023). However, due to the sparse gradient problem, it is difficult to add additional heads with randomly initialised parameters as they always underperform the pre-trained heads. Finally, MCL does not scale well to predicting larger numbers of outputs, due to poor data efficiency, as each head will on average only be exposed to the inverse percentage of the number of heads in the model, i.e. with 10 heads, each head on average will receive gradients from 10% of the dataset.

Finally, there are concerns over how flexibly MCL can be applied to different domains. Most MCL literature has focused on computer vision tasks (Guzmán-

Rivera et al., 2012; Kirillov et al., 2015; Lee et al., 2016; Firman et al., 2018). However, to the best of our knowledge, we are unaware of published literature applying the MCL architecture to NLP tasks. Indeed, for a task such as text generation, where the final layer of the model is the vocabulary layer, MCL would seemingly require running multiple copies of this vocabulary layer, significantly increasing the parameter count.

As a result of these issues, there is critical need for research to create more fundamental and flexible solutions to the one-to-many problem which are readily adaptable to multiple domains. It is this challenge that the remaining chapters of the thesis seek to address.

2.10 Evaluation

In this section, we provide an overview of popular evaluation methods for the tasks covered in this thesis. We first explain the common evaluation methods used for each modality and then show how they can be adapted towards the one-to-many variants of these tasks, as well as critical differences between tasks depending on what they are trying to achieve. In text-generation-based tasks, it is typical to use term-level overlap metrics. These are part of a class of evaluation metrics called 'automated' metrics, as they can be computed deterministically using pre-defined routines, in contrast to human-based metrics which require a human to evaluate the generated text. Overall, the advantage of automated metrics is that they are more reproducible between different research labs. By contrast, human evaluation may be influenced by subtle factors such as the amount paid to each human worker, the intructions they are given, or even the time of day, leading to workers having a different geographic distribution (the acquisition of workers is often outsourced to platforms such as Amazon Mechanical Turk). A core disadvantage however is the inflexibility of the evaluation. Typically, they rely on simple term-overlap-based methods, by comparing term-level similarity between the prediction and the ground-truth target. For instance, if the prediction is `I am fine thanks` and the ground-truth is `I am good`, then splitting these sentences into individual words, one can see that the prediction is somewhat close to the ground truth, with two words shared between the two. Typically, we evaluate the similarity by splitting the sentence into individual words (unigrams), however any arbitrary n-gram is possible such as bigrams, which would result in the tuples: {(I, am), (am, fine), (fine, thanks)}. These higher-order n-grams are sometimes used because these evaluation metrics are usually word-order agnostic (e.g. `am thanks fine I` is equivalent to the above), hence considering local words to each word helps to softly preserve some sense of word order. Below, we give more concrete examples of how specific automated metrics.

2.10.1 F1 score

Suppose from the above example, the ground-truth had actually been `I am fine`; every word in the prediction is contained in the ground-truth, but there is also a redundant word 'thanks'. Should we assign a 100% accuracy here, or should there be a penalty for the redundant word? F1 score addresses this by leveraging two popular metrics: *precision* and *recall*. Precision represents the proportion of predicted items that are contained in the ground-truth; In this case, 75%. Recall, represents the proportion of items in the ground-truth contained in the prediction; In this case, 100%. F1 score averages these two quantities via a harmonic mean:

$$F1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2.17}$$

Resulting in an F1 score of $\approx 85\%$. Note that the harmonic mean is used over a traditional arithmetic mean as it penalises imbalances between precision and recall more. This is helpful as in many tasks either precision or recall may be much easier than the other. E.g., when predicting cancer where 99% of the dataset contains healthy subjects, always predicting healthy enables 99% recall, but 0% precision and 0% F1 score.

2.10.2 ROUGE

ROUGE (Lin, 2004) extends the framework of F1 score by providing a suite of metrics. This thesis utilises the following three from that suite: ROUGE-1, ROUGE-2 and ROUGE-L. The former two are just the unigram and bigram variants of F1-score. However, as these are largely word order invariant, beyond the word adjacency of bigrams, ROUGE-L is also used to capture the longest common subsequence between the ground-truth and prediction, in this case `I am fine`.

2.10.3 Limitations

A core limitation of these term-based methods is that they are a hangover from tasks such as machine translation and information retrieval. In translation, although there are multiple valid translations of a given sentence, those multiple translations often still contain many overlapping words, e.g. in French, the presence of the word 'je' heavily implies the word 'I' will feature in the translation. Similarly in information retrieval, if I submit the query 'house for sale in Nottingham', documents containing the words 'house', 'sale' or 'Nottingham' have a decent chance of being relevant. These tasks contrast with their usage in more open-domain tasks such as dialogue, where the range of possible responses to a given message is vastly larger. Hence, if the ground-truth is `I am fine`, then `Doing pretty well` is also an acceptable response, yet has a pre-

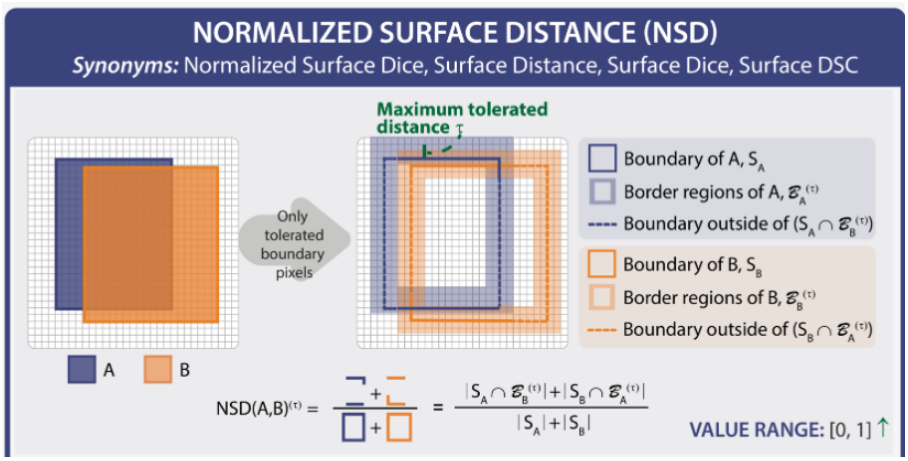


Figure 2.2: Graphical representation of normalised surface distance. Credit: https://metrics-reloaded.dkfz.de/metric?id=normalized_surface_distance

cision, recall and F1 score of 0%. As such, one typically observes much lower term-level scores in dialogue tasks, often in the low single digits, depending on the dataset (Li et al., 2016a), while machine translation tasks can easily achieve 30%+ (Vaswani et al., 2017b).

2.10.4 Image Segmentation Metrics

Image segmentation evaluation metrics have only partial overlap with those used in text generation tasks. Here, automated metrics are popular and to an extent they lack the limitations seen in text generation. When comparing a predicted segmentation with a ground-truth segmentation there are broadly two approaches that can be taken: overlap-based methods and edge-based methods. The overlap-based methods are most similar to text-based methods like F1 score and they consider the pixel-level overlap between the 2D annotations, such as DICE score (2.3). However, it can lack fine-grained sensitivity to differences in the edges, particularly if the object being annotated has a large surface-area relative to its circumference. As such, edge-based metrics consider the distance between the contours of the ground truth and prediction, without paying attention to internal pixels. One example of this that we use is normalised surface distance (NSD) (Maier-Hein et al., 2022) (Figure 2.2), which looks only at the overlap between pixels that fall within a certain distance to the boundary of the ground truth and prediction. In the chapters that follow, we use both metrics to capture different aspects of the segmentation task.

Part I

Implicitly One-to-Many

Chapter 3

Mode Unavailability

3.1 Introduction

This chapter addresses the first part of mode mismatch problem introduced in Section 1.1.1. As previously described, there are two aspects to this which are almost like two sides of the same coin–mode unavailability, which is addressed in this chapter and mode collapse, which we address in the subsequent Chapter 4. While later chapters in this thesis will focused on more downstream tasks such as how to optimise producing multiple outputs and how to roll out these techniques across multiple modalities, these first two chapters can be seen as laying the groundwork to address critical structural relationships between how the task, dataset and model interact with each other, that leads to one-to-manyness within the problem.

Let us further examine the case of mode unavailability, which is the problem this chapter tackles. This is a significant obstacle to the one-to-many problem, because even if we have a model capable of predicting multiple outputs, it will fail when it tries to predict an output that does not exist. For, many machine learning

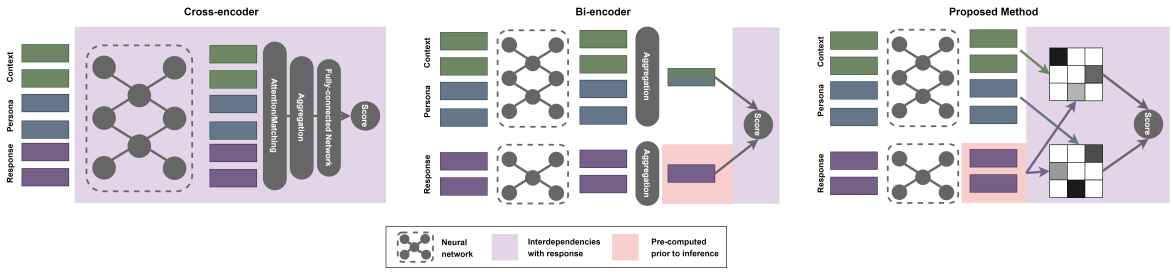


Figure 3.1: Comparison between cross-encoders, bi-encoders and our proposed method in personalised dialogue. Like bi-encoders, our method separately encodes responses to keep inference fast, but performs matching at the term-level and weights persona profiles according to their posterior relevance with the response.

Persona profiles	I have two dogs and one cat I work as a nurses aide in a nursing home I love to ride my bike I love caring for people
Context	Very cool. Do you have pets? No I do not, do you? Yes. Two dogs and a cat. they are my babies. Nice , where do you live ? I resign [sic] in North Dakota
BERT Bi-Encoder	That is so cool. I am a nurses aide at the old folks home
BERT-PAROR (ours)	I live in texas. I love riding my bike here.

Table 3.1: Sample from the PersonaChat dataset (Zhang et al., 2018a). Colour-highlights indicate semantic overlap. Through posterior term-level matching PAROR selects a response that takes both the context and most relevant profile into account, whereas the baseline bi-encoder only trivially repeats one of the profiles.

tasks involve selecting an output from a *constrained* list of outputs. In dialogue modeling, this is the difference between a generative model that generates tokens one by one and a retrieval model that selects a reply from a predefined list of candidate replies (Lowe et al., 2015; Kadlec et al., 2015; Wu et al., 2017; Zhou et al., 2018; Tao et al., 2019; Gu et al., 2019a; Yuan et al., 2019). This type of retrieval-based model is often preferred in cases where the types of replies a model can give need to be guardrailed. However, dialogue is inherently one-to-many – for a given dialogue context, there are innumerable plausible replies.

Overall then, our philosophy for managing this problem can be summarised as a look-before-you-leap approach, where we first want to be aware of which of the plausible replies are available as options, before we decide which to select. To give a simple example of how this problem might look in practice, note that most retrieval problems represent the things to be retrieved as mathematical vectors. The query, then, is also represented as a vector, and the nearest thing in vector-space is returned. E.g. if the query is 'what is your favourite animal?', suppose the model tries to predict the vector with the meaning *my favourite animal* in order to predict 'aardvark', however, because the database of animals we are working with is relatively small, it doesn't contain 'aardvark'. Instead, it returns whichever vector is closest to aardvark, perhaps a 'shrew'. However, the next nearest vector to aardvark may be quite far away and as such there is no guarantee it will be anything close to the user's favourite animal. If we had known aardvark didn't exist in the database however, we could have instead predicted the vector *my second favourite animal*, which may exist in a very different part of the vector space, and would provide a more relevant search result.

We approach this first from the perspective of persona-grounded dialogue, a task first introduced by the CONVAI2 competition (Zhang et al., 2018a). Specifically, this task combines aspects of both document retrieval and dialogue. To create the dataset, crowdworkers were randomly paired up and each assigned a 'persona' comprising a few short descriptive sentences termed *profiles* (Zhang et al., 2018a; Li et al., 2016c; Mazaré et al., 2018a). They were then instructed to get

to know one another. Given the dialogue context (the sequence of previous utterances in the conversation) and the speaker's persona, the goal is to predict which reply the human made, out of a pool of the correct reply and some distractor incorrect replies. The idea behind this challenge is to test whether an AI can maintain a consistent persona in a dialogue, rather than its persona simply being the amalgamation of all speakers in the dataset it was trained on. As shown in Table 3.1, in addition to just responding to the question 'where do you live?' posed by the speaker, a personalised chatbot can inject additional relevant persona information such as 'I love riding my bike here'

However, because of the multifaceted nature of this persona, depending on which profile the speaker chooses to emphasise, the speaker's reply will be different and it is usually difficult to predict this from the context alone (Kim et al., 2020; Lian et al., 2019). Currently, there are broadly two categories of retrieval models, both of which fail to adequately deal with this issue. (1) Bi-encoders convert the ⟨context,persona⟩ into a vector representation, which is then compared to the candidate responses – themselves also as vectors – using dot product similarity, with the highest scoring candidate being retrieved. However, as the bi-encoder does not know in advance what candidates are available the vector representation of the context and persona is likely to contain redundant information, i.e. it may try to predict a reply that does not actually exist in the candidate pool; (2) Cross-encoders side-step this issue by applying much more granular attention between the context, persona and response candidates. However, because of this, the vector representations for the replies cannot be pre-computed, making them impossibly slow in deployment (Humeau et al., 2020a; Tahami et al., 2020).

Our novel bi-encoder method **PAROR** (**P**osterior-**A**ugmented **R**etrieval with **O**ffline **R**epresentations) overcomes this through directly using the attention scores from contextualised *term-level* representations of context, profiles and response to score each candidate with profiles weighted according to their *posterior* relevance with the response. To show the versatility of our method, we implement both BiLSTM and BERT-based (Devlin et al., 2019a) versions of our model. Crucially, by keeping the ⟨context,profiles⟩ and response representations independent, we can entirely pre-compute the response representations offline, allowing for extremely fast inference. Quantitatively, we evaluate our model against the state-of-the-art (SoTA) bi-encoder. For *R@1* accuracy, we obtain up to 46.1% and 68.4% improvement over the SoTA bi-encoder in the PersonaChat original and revised datasets respectively (Zhang et al., 2018a). Qualitatively, as Table 3.1 shows, our model is able to produce a response that takes *both* the context and most relevant profile into account, whereas the standard bi-encoder only trivially repeats one of the profiles.

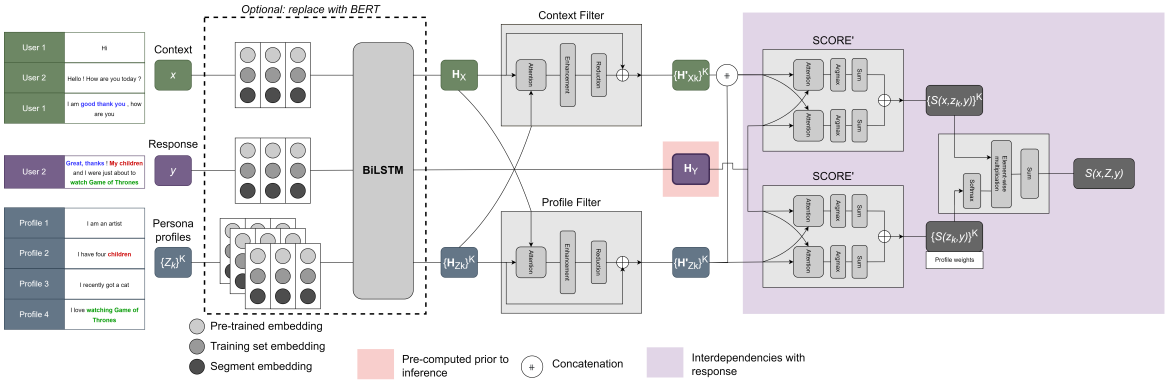


Figure 3.2: Overview of the proposed PAROR model. Colour-highlights in text (left) indicate semantic overlap.

3.2 PAROR

Figure 3.2 overviews our method: We separately encode context, persona and response before augmenting context and persona representations with a filtering layer. Our key novelty is the introduction of two late-stage interaction modules: a posterior ⟨profile, response⟩ scoring which then provides persona weights for the ⟨context, profile, response⟩ scoring. Crucially, the model decouples response and ⟨context, persona⟩ representations, allowing for responses to be precomputed prior to inference.

3.2.1 Task Definition

We define a sample in a dataset $\mathcal{D}$ as the tuple $\langle x,Z,y\rangle$ shown in Figure 3.2; x is the context; Z is the persona, consisting of a set of K profiles $\{z_k\}_{k=1}^{K}$; y is the response. $S(x, Z, y)$ is the similarity score between $\langle x, Z\rangle$ and y.

3.2.2 Word Embedding & Encoding

We use the word embeddings provided by (Gu et al., 2020b), combining 300-dim general pretrained GloVe embeddings (Pennington et al., 2014b), with 100-dim task-specific embeddings obtained via the word2vec algorithm on the training data (Mikolov et al., 2013). We also learn 4 64-dim segment embeddings during end-to-end training for each of the response, persona, context speaker A, and context speaker B (Gu et al., 2020a; Bao et al., 2020). This improves term-weighting in the scoring module (Section 3.2.4), as each segment may require different scaling. We define the concatenation of these embeddings along the hidden-state as $\mathbf{X}$, $\mathbf{Z}_k$ and $\mathbf{Y}$ for the context, persona and response respectively. We encode each separately with a BiLSTM (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) with shared weights to obtain the hidden states

$\mathbf{H}_x$, $\mathbf{H}_{z,k}$ and $\mathbf{H}_y$. We additionally experiment with replacing the above GloVE + BiLSTM method with the pretrained BERT model, which has demonstrated strong performance in a range of NLP tasks (Devlin et al., 2019a). We refer to this method as BERT-PAROR.

3.2.3 Filtering

We next apply the filtering layer adapted from (Gu et al., 2020b) to both the context and profiles. We apply this to each profile *separately*, which obtains K different views of the context where K is the number of profiles. This mitigates the one-to-many problem between contexts and responses (Zhao et al., 2017b). Filtering the profiles with the context helps to filter any profiles with a high response matching score, but which are not appropriate to introduce at the current point in the dialogue. Formally, we perform attention (Vaswani et al., 2017a) with the context $\mathbf{Hx}_x$ as queries and the k-th profile $\mathbf{H}_{z,k}$ as keys, followed by softmax normalisation to obtain the outputs $\tilde{\mathbf{H}}_\mathbf{x}$. We then perform an enhancement operation on each token via the following concatenation (Mou et al., 2016):

$$\hat{\mathbf{H}}_x = [\mathbf{H}_x; \tilde{\mathbf{H}}_x; \mathbf{H}_x - \tilde{\mathbf{H}}_x; \mathbf{H}_x \odot \tilde{\mathbf{H}}_x] \tag{3.1}$$

Finally, we perform a reduction operation to reduce each hidden state to its original size with a feed-forward network followed by a ReLU activation function (Nair and Hinton, 2010) and a skip-connection.

3.2.4 Scoring

Our scoring procedure is the key novelty of our method. Unlike previous bi-encoders which aggregate ⟨context,persona⟩ and response representations into single vectors (Zhang et al., 2018a; Mazaré et al., 2018a), we allow for weighted term-level matching between context, persona and response representations, to capture fine-grained semantic similarity. Crucially however, by delaying this operation until the very end, we maintain the inference efficiency of bi-encoders, as the response representations $\mathbf{H}_\mathbf{y}$ can be pre-computed prior to inference. Our scoring procedure is inspired by late-stage interaction framework of ColBERT for the openQA and IR tasks (Khattab et al., 2021; Santhanam et al., 2021). Specifically, given a similarity matrix $M \in \mathbb{R}^{I \times J}$ between query terms $Q \in \mathbb{R}^{I \times D}$ and key terms $K \in \mathbb{R}^{J \times D}$, for each term in the query we take the argmax and sum them together to obtain a final score between the overall query and key. We define this function as $SCORE(Q, K)$. We make several significant novel changes detailed below to adapt this for personalised dialogue.

Term Similarity Metric We use unnormalised dot product scoring, rather than cosine similarity or L2 distance, to compute similarity between terms, allowing implicit term-weights to be learned, similar to sparse retrieval (Ramos, 2003; Robertson and Zaragoza, 2009). This is important as query length in dialogue is much longer than in openQA/IR, and thus contains many irrelevant terms which need to be filtered out.

Bi-directional Scoring Unlike documents in openQA/IR which have a much flatter distribution, certain responses occur very frequently in many contexts. Therefore, we apply the SCORE function bi-directionally with a 50/50 weighting (i.e. $BiSCORE(Q, K) = 0.5 \cdot SCORE(Q, K) + 0.5 \cdot SCORE(K, Q)$, to prevent the model defaulting to frequently occurring responses (these will receive a low $SCORE(K, Q)$ result) (Li et al., 2016b).

Posterior persona Selection Selecting a profile using the context is often difficult, due to limited semantic overlap between context and profiles and the one-to-many relation between them. By contrast, selecting profiles using the response is often trivial (Kim et al., 2020; Lian et al., 2019). We therefore score each against the response using the $BiSCORE$ function to obtain $\{S(z_k, y)\}_{k=1}^{K}$. This boosts informative profiles, but diminishes uninformative ones.

Response Scoring We next concatenate the context and profile representations length-wise and score them against the response representations using $BiSCORE$, to obtain a separate score for each $\langle$context, profile$\rangle$ pair $\{S(x, z_k, y)\}_{k=1}^{K}$. Finally, we obtain the weighted average of these scores by multiplying each score by its corresponding softmax normalised $\langle$profile, response$\rangle$ score obtained in Section 3.2.4 and summing over the outputs:

$$S(x, Z, y) = \sum_{k=1}^{K} [S(x, z_k, y) \cdot \frac{\exp(S(z_k, y))}{\sum_{k'}^{K} \exp(S(z_{k'}, y)}]$$

Loss Function Following previous work (Mazaré et al., 2018a; Zhang et al., 2018a) we train the model with cross-entropy loss (Section 2.3), using the other samples in the batch as negatives (i.e. the logits consist of a matrix $\mathbf{M} \in \mathbb{R}^{I \times I}$ where I is the batch-size, and $M_{i,j}$ is the score $S(x^i, Z^i, y^j)$ between the i-th $\langle$context, profiles$\rangle$ and the j-th response).

3.3 Experiment

3.3.1 Experimental Setup

Dataset We test our method on the PersonaChat dataset (Zhang et al., 2018a), which is a persona-grounded dataset where each speaker is assigned a persona consisting of 3-5 short sentences termed *profiles*. The dataset also provides more challenging *revised* personas, which contain precisifications or generalisations of the *original* personas: e.g. 'I like playing sports' could become 'I play football every weekend' or vice versa. Note, models are trained *separately* on these tasks. The dataset contains 8939, 1000 and 968 dialogues and 65719, 7801 and 7512 samples for training, validation and testing respectively.

Training Details We train using the Adam optimizer (Kingma and Ba, 2015) with a learning-rate of $1e-3$, with decay of 0.8 every epoch for 10 epochs, followed by decay of 0.2 for 4 epochs. For BERT-PAROR, we use a learning-rate of $5e-5$, with decay of 0.4 when validation loss plateaus (Humeau et al., 2020a). We use a batch size of 128 for PAROR and 32 for BERT-PAROR. For PAROR only: We use a hidden state size of 200 in the LSTM with 2 layers, gradient norm clipping of 1.0, dropout of 0.2 in the word embeddings and hidden layers (Srivastava et al., 2014) and label smoothing (Szegedy et al., 2016) of $\epsilon = 0.05$ to reduce overfitting (Pereyra et al., 2017). We concatenate the last 4 turns to obtain the context x for both models.

Baselines We compare our method to all the baselines from the latest work on personalised dialogue (Gu et al., 2020b; Zhang et al., 2021a; Hua et al., 2020). While our core comparison is against other bi-encoders due to their comparable latency, the cross-encoders provide a useful indicator of an upper bound to performance. DGMN (Zhao et al., 2019b), MNDB (Zhang et al., 2021a), DIM (Gu et al., 2019b), CSN (Zhu et al., 2021), RSM DCK (Hua et al., 2020), FIRE (Gu et al., 2020b) and BERT CRA (Gu et al., 2021) all implement various forms of cross-attention between context, persona and response. TransferTransfo (Wolf et al., 2019) and P^2 Bot (Liu et al., 2020) are generative language models that multitask on the retrieval task. IR baseline, Starspace, Prof. Mem. and KV Prof. Mem. were introduced in the original PersonaChat paper (Zhang et al., 2018a). Transformer (Mazaré et al., 2018a) is the current SoTA bi-encoder, which is a memory network (Sukhbaatar et al., 2015) with a transformer encoder (Vaswani et al., 2017a). We also run two additional baselines: MemNet is our memory network implementation (see Table 3.3); BERT Bi-Encoder adopts the method of (Ouguz et al., 2021) which is the current SoTA with pretrained models. This involves concatenating the entire persona to the context as one long string and then separately encoding the $\langle$context, persona$\rangle$ and response and scoring with the dot product of their aggregated vector representations. Due to computing and resource limitations, we are not able to achieve as high as the numbers reported

Type	Model	$R@1$	
		Original	Revised
Cross-encoders	DGMN (Zhao et al., 2019b)	67.6	58.8
	MNDB (Zhang et al., 2021a)	75.6	73.6
	DIM (Gu et al., 2019b)	78.8	70.7
	CSN (Zhu et al., 2021)	78.1	71.3
	RSM-DCK (Hua et al., 2020)	79.65	71.85
	TransferTransfo (Wolf et al., 2019) †	80.7	–
	P^2 Bot (Liu et al., 2020) †	81.6	74.8
	FIRE (Gu et al., 2020b)	**81.6**	**74.8**
	BERT-CRA (Gu et al., 2021) †	**84.3**	**79.4**
Bi-encoders	IR baseline (Zhang et al., 2018a) ‡	41.0	20.7
	Starspace (Wu et al., 2018)	49.1	32.2
	Prof. Mem. (Zhang et al., 2018a)	50.9	35.4
	KV Prof. Mem. (Zhang et al., 2018a)	51.1	35.1
	Transformer (Mazaré et al., 2018a)	54.2	42.1
	MemNet ‡	60.5	57.0
	BERT Bi-Encoder †‡	76.1	69.4
	PAROR (Ours) ‡	**79.2**	**70.9**
	BERT-PAROR (Ours) †‡	**80.6**	**75.3**

Table 3.2: Results of PAROR and BERT-PAROR and previous methods on the PersonaChat test sets. †= uses pretrained model. ‡= results obtained from our run.

in Ouguz et al. (2021). As we train on only a single GPU, we use a smaller batch size / context window, share parameters between the encoders and do not use additional training data beyond the BERT checkpoint. However, this makes our training hyperparameter setup identical to BERT-PAROR, allowing for fair comparison. In principle however, similar scaling techniques as employed in (Ouguz et al., 2021) could be applied to BERT-PAROR if more compute / memory were available.

Evaluation Both bi-encoders and cross-encoders are evaluated using the standard IR metric *Recall@1 (R@1)*, which measures each model's ability to select the gold response from a pool of 20 candidates containing 19 random negatives (Dinan et al., 2019).

3.3.2 Experimental Results

Main Results As shown in Table 3.2, our method sets a new SoTA for bi-encoders for both tasks, with PAROR surpassing Transformer (Mazaré et al., 2018a) in *R@1* by 46.1% and 68.4% in the PersonaChat original and revised benchmarks respectively and BERT-PAROR surpassing BERT Bi-Encoder by

5.9% and 8.5%. Both models are statistically significantly better on a t-test with p-value < 0.05 than the next best models MemNet and BERT Bi-Encoder respectively. Unsurprisingly, the SoTA cross-encoders FIRE (Gu et al., 2020b) and BERT-CRA (Gu et al., 2021) perform slightly stronger, due to the ability to perform fine-grained deep interaction between context, profiles and response; however, PAROR retains 97.1% and 94.8% of the performance in the respective tasks, and similarly 95.6% and 94.8% for BERT-PAROR. This suggests the key performance improvement is driven by the removal of the single-vector bottleneck of bi-encoders (see Section 3.3.2). Our method also outperforms in the revised dataset, which has typically been difficult for bi-encoders, due to the vocabulary mismatch between profiles and responses (see Section 3.3.1): e.g. the Transformer only retains 77.7% of its performance versus the original dataset, versus 89.5% for PAROR, which is much closer in-line with cross-encoders levels (e.g. 89.7% for DIM; 91.7% for FIRE). To validate the efficiency of our precomputation strategy we record the time it takes for PAROR versus the SoTA cross-encoder FIRE (Gu et al., 2020b) to retrieve a response from the entire test set (7512 candidates) using an NVIDIA P100 GPU. Our model takes just 3ms versus 3570ms for FIRE.

Ablation In Table 3.3, we present ablation studies removing key components from PAROR. Note each ablation is conducted separately rather than cumulatively. We perform the following six ablations: (A) We replace the $BiSCORE$ operation with a Max Pool aggregation over the hidden states, so that each context, profile and response is represented as a single vector, similar to previous bi-encoder methods (Mazaré et al., 2018a; Zhang et al., 2018a); (B) We use the context, rather than the response to weight the profiles; (C) We replace the dot product with cosine similarity for the $BiSCORE$ operation; (D) We use a unidirectional $SCORE$ function (see Section 3.2.4); (E) We remove the segment embeddings; (F) We remove the filtering layer. We find all ablations harm performance, especially (A) and (C) which are part of the key novelties to our work, confirming the effectiveness of our method (statistically significant on t-test with p-value < 0.05). We also combined ablations (A), (B), (E) and (F) which essentially results in a memory network (MemNet) similar to previous methods (Mazaré et al., 2018a). Lastly we removed the persona from PAROR entirely, which significantly harmed performance, demonstrating the importance of having a method which can effectively use the available profiles.

3.4 Conclusion

In this chapter we presented a novel method **PAROR** (**P**osterior **A**ugmented **R**etrieval with **O**ffline **R**epresentations). We showed how this approach could sidestep the mode unavailability problem caused by the mismatch between plausible outputs and available outputs and overall improved model performance as a result resulting in up to 68.4% outperformance over the SoTA bi-encoder meth-

Model Configuration	$R@1$
PAROR	**79.2**
A: Replace MaxSim with Max Pool	72.9
B: Score persona using context	78.7
C: Replace dot product with cosine similarity	75.6
D: Unidirectional $SCORE$	78.2
E: No segment embedding	78.2
F: No filters	78.0
A + B + E + F (MemNet)	60.5
No persona	63.6

Table 3.3: Results of PAROR and various ablations on the PersonaChat original test set. Ablations are applied separately, rather than cumulatively (see 3.3.2 for details).

ods.

As a result, this removes a core restriction with one-to-many models. Namely, artificial constraints preventing them from demonstrating the full range of possible outputs, even if the model was hypothetically capable of doing so. Recall that model unavailability was only one half of the originally introduced problem of mode mismatch in Section 1.1.1. In the next chapter, we will address the other half, namely mode collapse.

Chapter 4

Mode Collapse

4.1 Introduction

In Chapter 3, we addressed the first half of the mode mismatch problem introduced in 1.1.1–mode unavailability, i.e. when there are multiple plausible outputs to the task, but only a handful are available for selection (e.g. because there is a fixed pool of outputs) how do we prevent the model from predicting something that doesn't exist, and ultimately enable more relevant predictions through this awareness. However, this can be seen as trying to solve an inference-level problem, i.e. we have a model with a one-to-many capability, it knows the full range of outputs given an input, but how do we ensure that it limits itself to available outputs when the domain is constrained. What we want to focus on now though is the earlier problem of mode collapse which occurs in training. To reiterate, mode collapse occurs where the ground-truth distribution has multiple modes, e.g. for the distribution of the weight of each member of a group of animals comprising multiple different species, we might expect one mode per species. In practice, for complex tasks such as language, there is usually a long tail distribution over these modes (see Section 2.3). As such the model often fails to learn the full range of this distribution, stopping short and only learning the bulkier 'head and body' of the distribution. In other words, it only learns part

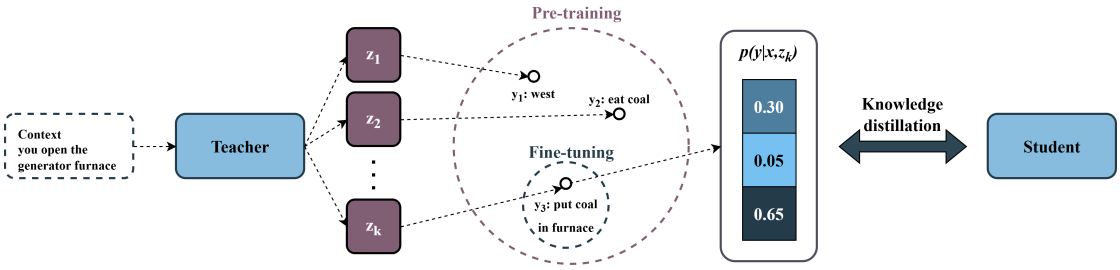


Figure 4.1: Overview for PASA. A teacher model learns to predict K different actions to capture the pre-training data distribution, and the posterior probability from the fine-tuning data is distilled into a student model.

of the one-to-many problem that is presented. This is a problem in cases where the particular modes that we want the model to learn exist only in the tail of the distribution.

For instance, in dialogue there are many valid responses for a given context. However, only some of these responses will be desirable in our downstream task, while others will be counterproductive. For instance, dialogue models intended to engage the user in friendly chitchat are pre-trained on Reddit (Zhang et al., 2020), which contains a mixture of users who are thoughtful and interesting as well as hostile and argumentative (Xu et al., 2020). This raises the question of how we should train the model such that it can emulate the desirable behaviours on the downstream task, but not the undesirable ones.

The current predominant approach is to train in a one-to-one setup—only a single target response is given for a single dialogue context (Roller et al., 2021; Wolf et al., 2019). However, this leads to models only learning to predict the *average* response, while ignoring the full range of *possible* responses (Li et al., 2016a).

To explore solutions we leverage the domain of text-based games (Liu et al., 2023a). Text-based games are a form of human-AI interaction in which an AI provides a description of a textual world, and a human can interact with this by providing natural language commands ('actions')—which are interpreted by the game's hard-coded parser—and receiving textual observations from the game's interface (see Figure 4.1). The objective is to maximise the integer score by completing various in-game tasks such as exploring new areas, collecting valuable items or solving puzzles. Text-based games are especially well-suited to the task for two key reasons:

- There are two relevant datasets—ClubFloyd (Yao et al., 2020) and Jericho Walkthroughs (Liu et al., 2023a)—which mirror the aforementioned situation in dialogue. ClubFloyd is a comparatively large dataset of humans playing text-based games, and therefore contains a mixture of behaviours, ranging from expert humans who obtain the maximum score, to novice humans who obtain virtually no score, which parallels the mixture of friendly and unfriendly dialogue behaviours in the aforementioned Reddit example. Jericho Walkthroughs is a much smaller dataset containing only the exact actions required to obtain the maximum score. Thus, ClubFloyd represents our pre-training dataset containing a mixture of desirable and undesirable behaviour and Jericho Walkthroughs represents our fine-tuning dataset containing only the desirable behaviour.
- Through backend access to the game engine, we can train and evaluate directly with the set of valid actions—those actions that are understood by the game's parser given the current game state. Unlike dialogue, where most datasets do not contain the set of valid responses, this allows us to explicitly evaluate the task of predicting the best (sc. *gold*) action—the action that maximises long-term score—given the set of valid actions, which forms the key focus for this chapter.

Our proposed approach, PASA (**P**osterior **A**ction **S**election from Discrete L**a**tent Variables), uses discrete latent variables to capture the range of different behaviours represented in our larger pre-training dataset. Particularly, we learn to predict a set of possible actions, where each action reflects a different value for the latent variable. We assign our latent variable using the ground-truth action itself, via a posterior recognition network. Unlike previous approaches, we do not then learn to predict the latent variable from the context. This is because our goal is not to learn to output diverse responses as is the goal of previous approaches (Zhao et al., 2017a), but rather to model the full set of possible responses. Then, the second novelty in this chapter lies in our method for re-transforming what our model has learned into a one-to-one setup. Given our teacher model which predicts sets of actions, we use knowledge distillation (Hinton et al., 2015b) to distill the posterior probability distribution into a student model which predicts the gold action given the context, when fine-tuning on the Jericho Walkthroughs dataset. This probability distribution is far richer than learning from only the hard targets of the dataset, and thus allows the student model to benefit from the richer range of actions the teacher model has learned. As shown in Figure 4.1, during pre-training we learn the set of possible actions from a given context, which ranges from poor actions ('eat coal'), to average actions ('west') to high-quality actions ('put coal in furnace'). Then during fine-tuning we specialise the model to only predict the high-quality actions.

4.2 Method

In this section, we describe our PASA architecture (see Figure 4.2), which comprises a teacher model which uses posterior latent variables, and a student model, which is trained via knowledge distillation on the soft targets of the teacher model (Section 2.7). Both models are retrieval models, in that they aim to select the gold action, given a subset of actions.

4.2.1 Task definition

We focus on the problem of choosing the gold action in text-based games, as defined by a set of walkthroughs which comprise the sequence of actions required to maximise the score, from the set of valid actions provided by back-end access to the game engine. Formally, we learn to predict the action y_t at time t, from the current context x_t, which is given as the last T elements from the current trajectory of interwoven actions and observations since the start of the game $< o_1, y_1, ..., y_{t-1}, o_{t-1} >$.

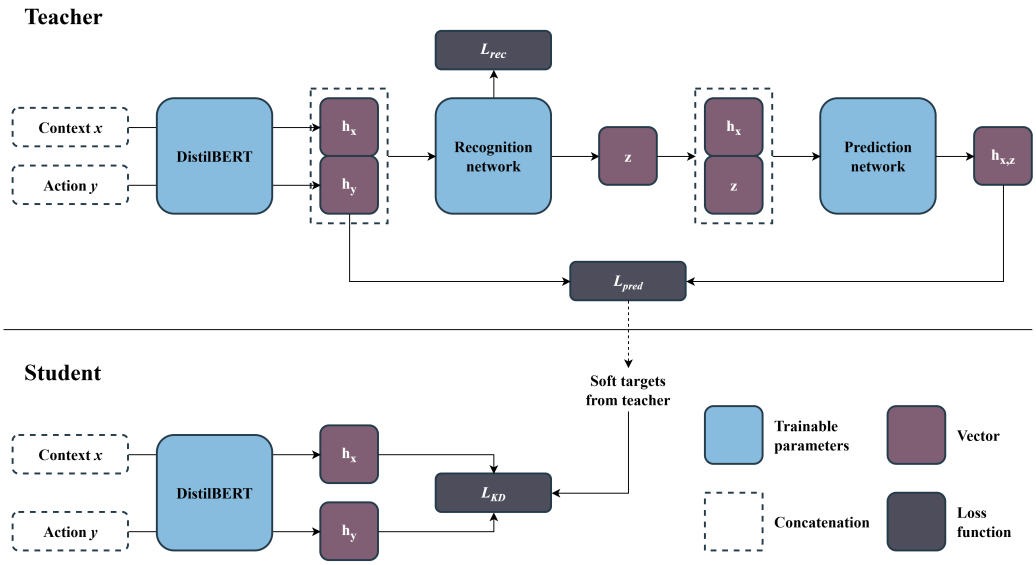


Figure 4.2: The architecture for PASA. The teacher model is trained to predict the next action, by being given access to posterior information z about the action. In this case, the posterior information reflects some hint about what action is being taken, such as the type of action (we explore different options here). From this, the teacher outputs a probability distribution over possible actions, called 'soft targets', as opposed to 'hard targets' where p=1 for the ground truth action and p=0 elsewhere. The student then learns to imitate these soft targets, providing a richer learning environment.

4.2.2 Encoding

For encoding both contexts and actions, we use the DistilBERT model (Sanh et al., 2019b), which is a 6-layer 66M parameter bidirectional transformer, trained via knowledge distillation (Hinton et al., 2015b) on BERT (Devlin et al., 2019b), a transformer pre-trained on Wikipedia and human written books (see Section 2.4). We *separately* encode the context and each action by inputting the sequence of tokens from each into the model, which return the hidden states for that sequence. In front of each sequence, we have pre-pended a `[CLS]` token, which serves as a vector-based representation of the overall sequence. We denote $\mathbf{h}_c$ and $\mathbf{h}_x$ as the vector output from DistilBERT for the context and action respectively.

4.2.3 Student

The student is the simpler of the two models, as it does not have the separate recognition and prediction networks of the teacher. While the teacher is essentially a type of CVAE (see Section 2.8), the student is ultimately just a bi-encoder, as described previously in Chapter 3. That is to say, after separately encoding both the context and action, the compatibility between these is represented as the

dot product similarity between these:

$$S(x_t, y_t) = \mathbf{h}_x \cdot \mathbf{h}_y \tag{4.1}$$

These scores can then be transformed into probabilities via a softmax normalisation.

4.2.4 Teacher

Open-ended language tasks such as dialogue or summarisation can be framed as one-to-many problems—i.e. there are many acceptable responses/summaries for a particular context. We can represent this by introducing a latent variable z, which is a single 1-of-K categorical variable which is also used to condition the action in addition to the context. Intuitively, this allows our model to predict K different actions from a single context, simply by fixing z to different values. To achieve this, we extend our student model with both a recognition network, which learns to predict the latent variable z and a prediction network, which synthesises both the context and chosen value for z.

Recognition network. The recognition network takes as its inputs both the context and action and outputs a posterior probability distribution over z. We concatenate the output vectors from each $\langle$context, action$\rangle$ pair and apply two fully connected layers followed by a softmax normalisation:

$$p(z|x, y) = \text{softmax}(\mathbf{W}_2(\mathbf{W}_1([\mathbf{h}_x; \mathbf{h}_y]))) \tag{4.2}$$

Where $[\cdot;\cdot]$ represents vector concatenation, $\mathbf{W}_1 \in \mathbb{R}^{D\times K}$ and $\mathbf{W}_2 \in \mathbb{R}^{2D\times D}$. We then sample from this distribution using the Gumbel-Softmax re-parametrization trick to obtain low-variance gradients (Jang et al., 2017):

$$\mathbf{z} = \text{GumbelSoftmax}(p(z|x, y)) \tag{4.3}$$

Where $\mathbf{z}$ is a differentiable K-dimensional one-hot vector.

Prediction network. The prediction network serves to predict the ground-truth action. It takes as its input the context and one-hot encoded latent variable. These are concatenated and put through a fully connected layer where $\mathbf{W}_3 \in \mathbb{R}^{(D+K)\times D}$:

$$\mathbf{h}_{x,z} = \mathbf{W}_3([\mathbf{h}_x; \mathbf{z}]) \tag{4.4}$$

4.2.5 Semantically Aligning the Latent Variable

In the standard setup for latent variable models the recognition network is reliant on the gradients from end-to-end training in order to learn a good representation for z. Intuitively however, we might be able to guide the recognition network by using some heuristics to determine how z should be assigned as an additional loss function $\mathcal{L}_{rec}$ which acts as a soft constraint. We present three different approaches for this:

Latent. This represents the vanilla discrete latent variable approach, in so far as we do not explicitly tie the latent variable to any particular semantics. Instead, to encourage the model to make use of the full capacity of the latent variable, we regularise z by using a uniform prior with batch prior regularisation (BPR) (Zhao et al., 2018). Particularly, we take the approximated marginal distribution $q'(z)$ over the latent variable:

$$q(z) \approx \frac{1}{N} \sum_{n}^{N} q(z|x_n, y_n) = q'(z) \tag{4.5}$$

Then, we minimise the Kullback-Leibler (KL) divergence between the uniform prior $p(z)$ and our approximated marginal $q'(z)$.

$$\mathcal{L}_{rec} = \sum_{k}^{K} q'(z_k) \log \frac{q'(z_k)}{p(z_k)} \tag{4.6}$$

Note that this regularisation differs significantly from the more common approach of learning to predict a prior probability given the context and then directly minimising KL-divergence between the prior and posterior (Zhao et al., 2017a). Since, that approach is known to lead to posterior collapse, whereby the model outputs a fixed distribution for the latent variable regardless of the posterior information in order to trivially reduce the KL-divergence loss term. Despite not explicitly making the latent variable predictable from the context, our model should not learn to ignore the context as the capacity of the latent variable is highly constrained, and therefore the context is still required to accurately predict the action.

Persona. In a task such as dialogue, an intuitive answer to why different responses can be given to the same context is because there are different people responding, who may each have their own linguistic styles, attitudes and goals. Previous work on persona-grounding has therefore sought to inject speaker information into the context to aid prediction (Li et al., 2016d; Zhang et al., 2018a). Inspired by this, we treat each transcript from the pre-training data as a different

Intent	Matching Patterns	# Samples
navigate	north, east, south etc.	54.6k
examine	examine, look	39.1k
hoard	take, get, inventory etc.	18.5k
interact	contains verb and noun	25.9k
other	–	49.6k

Table 4.1: Rules for intent detection with number of samples by intent. Matching conditions are tested in the listed order. SpaCy is used for relevant part-of-speech tagging.

speaker, where each speaker represents a different value the latent variable z can take. We then add an additional loss term for the recognition network to predict the correct value for z given the context and action, which we train with cross-entropy loss. Note, we do not apply this loss function during fine-tuning as we have no labelled mapping between the speakers in the pre-training data and the creators of the walkthroughs.

Intent. In task-oriented dialogue, an intent represents a high-level action the user wishes to make, such as a greeting, asking a question, requesting information etc. (Young et al., 2013). Often, these intents can be identified at a coarse level using fairly simple regular expressions (He et al., 2018). We observe that many actions in text-based games can be hierarchically organised in a similar way. The current state-of-the-art agent for playing text-based games in zero-shot settings is a rules-based model NAIL (Hausknecht et al., 2019), which adopts a modular approach identifying four types of action: *navigation* - moving between locations; *examination* - using the examine and look commands to gather feedback about the environment's visuals; *hoarding* - accumulating as many objects as possible; *interacting* - interacting with the environment using objects from the inventory such as fighting a troll with a sword or opening a chest. As presented in Table 4.1, we use regular expressions and the SpaCy part-of-speech tagger [1] to classify actions according to this schema, and add an additional 'other' class for anything not matched. Then we train the recognition network to predict the correct class using cross-entropy loss.

4.2.6 Training pipeline

We train our models in four stages: (1) student pre-training, (2) teacher pre-training, (3) teacher fine-tuning, (4) teacher-student knowledge distillation. In each stage we optimise our model with cross-entropy loss to predict the ground truth action, given a selection of negatives:

[1] https://spacy.io/

$$\mathcal{L}_{pred} = -\log \frac{\exp(\mathbf{h}_x \cdot \mathbf{h}_y)}{\sum_j \exp(\mathbf{h}_{x,j} \cdot \mathbf{h}_{y,j})} \tag{4.7}$$

For pre-training using the ClubFloyd dataset, we use other actions from the batch as negatives, as we do not have access to the backend set of valid actions there. For stages (3) and (4), where we use the Jericho Walkthroughs dataset, we use the valid actions provided by the game engine as the negatives.

Knowledge distillation. In the standard setup for latent variable models it is common to train a prior network which must predict the latent variable given only the context (Zhao et al., 2017a). This is trained concurrently to the recognition network via reverse KL-divergence, which encourages mode-seeking behaviour from the prior network. However, this prevents our teacher model from learning to predict the full set of possible actions. As shown in Zhao et al. (2018), there is a conflict between trying to minimise the KL-divergence (important for the one-to-one task) and maximising the information between the actions and latent variable (important for the one-to-many task). Intuitively, to the extent that we can easily predict z from the context c, so too is z unlikely to tell us anything new about the action x that we didn't already know from c. We therefore approach transitioning from a one-to-many to a one-to-one task through knowledge distillation allowing us to keep the tasks fully disentangled. Particularly, given our discrete latent model fine-tuned on the Jericho Walkthroughs as the teacher Φ and the baseline model pre-trained on Clubfloyd as the student Θ. We then perform distillation on the Jericho Walkthroughs dataset. Specifically, we train the student model using a combination of the soft targets (see Figure 4.2) provided by the teacher network $p_\Phi(\cdot)$, and the ground-truth hard targets $p_\Psi(\cdot)$ from the dataset itself, weighted by a coefficient β (set to 1 during all of our experiments):

$$\mathcal{L}_{KD} = -p_\Phi(y|x) \cdot \log p_\Theta(y|x) - \beta p_\Psi(y|x) \cdot \log p_\Theta(y|x) \tag{4.8}$$

Following Hinton et al. (2015b), we apply a temperature of 20 to the logits for both models before the softmax operation for cross-entropy training on the soft targets only. This is because the teacher typically produces a very sharp probability distribution at a temperature of 1 - as it has already seen the training data - and therefore the student cannot learn much from the class probabilities.

4.2.7 Inference

During inference, we only use the student model, as the teacher model assumes access to the ground-truth action in order to perform posterior latent variable assignment. Intuitively, we hypothesise that the student model will perform better on the prediction task by having been trained on the richer probability distribu-

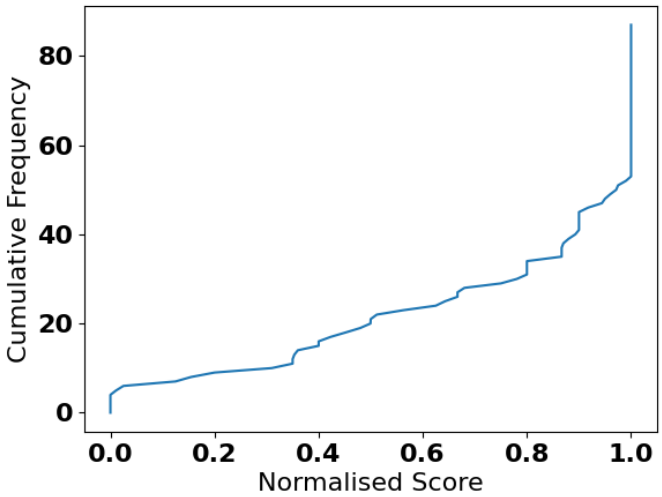


Figure 4.3: Cumulative frequency graph of normalised player scores from the ClubFloyd dataset.

	CF	JW
# Unique Games	590	28
Vocab size	39,670	9,623
Vocab size (game avg.)	2,263	1,037
Train samples	187,679	4,523
Test samples	22,947	1,626
Total samples	210,626	6,149
Avg. steps per reward	18.2	9.7

Table 4.2: Statistics for the ClubFloyd (CF) and Jericho Walkthroughs (JW) datasets.

tion from the teacher model, rather than the hard labels of the dataset.

4.3 Experiment

4.3.1 Datasets

The ClubFloyd (Yao et al., 2020) dataset comprises 426 transcripts of 590 unique games and 210,626 context-action pairs created through human gameplay [2]. We follow the same preprocessing approach as Yao et al. (2020): namely (1) we remove transcripts containing games in the Jericho platform, to ensure there is no contamination with the fine-tuning data; (2) We select 10% of the transcripts to form the validation data, noting that splitting at the transcript level prevents test information from leaking into the training data (e.g. an action appearing in

[2] https://github.com/princeton-nlp/calm-textgame

the context trajectory for a train sample being the target for a test sample).

We apply additional analysis beyond previous work to further understand the differences in action quality for ClubFloyd compared to a walkthrough containing only gold actions. Upon reaching a terminal state, most games will report the player's final score as well as the total possible score. Through pattern matching, we are able to extract final scores for 83 of the transcripts, which we normalise by dividing by the total possible score. We present these in Figure 4.3. As shown, there is significant variation in performance. Additionally, even the top scoring performances are far from being expert demonstrations, as these are only the final scores after players have completed large amounts of exploration, sometimes saving and loading between game states. Some games also report whenever the player's score changes, which we identify using regular expressions in 115 of the transcripts. In Table 4.2 we also report the average steps required per reward (only counting transcripts where at least a single reward is observed). We see that the walkthroughs require approximately half as many steps per reward, further validating the hypothesis that ClubFloyd contains suboptimal actions on average.

The Jericho Walkthroughs dataset contains 28 walkthroughs for the games supported by the Jericho platform [3]. Although these walkthroughs contain only the desired behaviour—i.e. gold actions—they are limited in number and therefore difficult to generalise from. Following previous work (Ammanabrolu and Riedl, 2021b), we split the train/test data by game, ensuring that the model is evaluated on unseen games. This is important to prevent contamination as the context for one sample also includes the ground truth action for another. However, it also adds an increased layer of difficulty as some games include unusual vocabulary, such as the name of a spell, which does not exist outside of the game. Unseen games therefore present a difficult generalisation task which requires significant language understanding. The games evaluated on are classified as either *difficult* or *possible* games, according to Liu et al. (2023a). This takes into account a variety of factors such as the size of the action space, how unusual the actions are (i.e. non-standard words such as the name of a spell), how sparse the reward is, and the complexity of the tasks required (easier games may reward the player for basic exploration, e.g. 'west' 'east' etc., while difficult games may require more reasoning such as moving a rug to reveal a trap door).

4.3.2 Experimental Setup

We train our models using the Adam optimizer (Kingma and Ba, 2015). We train for a single epoch during pre-training and 3 epochs during fine-tuning with a learning rate of $5e-5$. We use a batch size of 8. All of our methods use the DistilBERT model as their base, which provides a good trade-off between size and

[3] https://github.com/microsoft/jericho

	# Valid	CALM		PASA			Ablation
	Acts	PT	FT	Persona	Latent	Intent	Student-only
Difficult Games							
zork1	16.9	30.1	33.1	**53.0**	48.2	52.8	51.5
deephome	20.4	44.6	43.1	5.5	**40.1**	36.1	14.7
ludicorp	15.5	43.7	54.1	**70.6**	69.2	**70.6**	66.2
balances	24.2	29.5	28.7	54.9	**68.9**	64.8	49.2
Possible Games							
pentari	5.3	34.7	36.7	**44.9**	40.8	**44.9**	40.8
detective	8.1	60.8	47.1	52.9	**58.8**	56.9	52.9
library	8.6	26.9	26.9	**34.6**	28.8	30.8	32.7
temple	16.1	31.5	49.2	74.6	73.5	**79.6**	70.7
ztuu	34.8	29.8	33.3	21.4	**29.8**	26.2	27.4
Overall	17.8	37.1	41.6	47.5†	54.2†	**55.3**†‡	47.2

Table 4.3: Recall@k performance on the Jericho Walkthroughs test set. Games are classified as either 'possible' or 'difficult', using the classification from Liu et al. (2023a). †/‡ = statistically significant on t-test with p-value < 0.01 compared to CALM-FT and Student-only ablation respectively.

accuracy being 40% smaller than BERT, while retaining 97% of the performance across NLP tasks. We truncate contexts to the last three elements following Yao et al. (2020) such that $x_t =< o_{t-1}, y_{t-1}, o_t >$. During tokenization, we truncate the contexts and actions to a maximum of 128 and 8 tokens respectively. Following previous approaches (Ammanabrolu and Riedl, 2021b,a), we evaluate on the Jericho Walkthroughs test set, while our model is trained first on the ClubFloyd dataset, then fine-tuned on the Jericho Walkthroughs train set, following the procedure outlined in Sec 4.2.6. We evaluate using recall@1, which measures the model's ability to select the gold action from the pool of valid actions, which is a standard evaluation metric in retrieval tasks (Oğuz et al., 2021).

4.3.3 Baseline

For our baseline we use the CALM model (Yao et al., 2020) [4][5], which is a state-of-the-art approach to action prediction in text-based games and represents the paradigmatic one-to-one method by training GPT-2 (Radford et al., 2019b) to generate the next action given the context using the ClubFloyd dataset of human transcripts. Following their implementation, we score actions by summing over

[4] https://github.com/princeton-nlp/calm-textgame

[5] Another possible baseline candidate would have been the Worldformer (Ammanabrolu and Riedl, 2021a), however the code is not released and our evaluation task is different from the one they report on, therefore comparison is not possible.

the token-wise log-probabilities for each action. Additionally, we also further fine-tuned this model directly on the Jericho Walkthroughs dataset, to make for fair comparison with our approach. We refer to these approaches as CALM-PT and CALM-FT respectively.

4.3.4 Results

We report recall@1 performance on Jericho Walkthroughs in Table 4.3, with individual performance for each game, split according to the difficulty classification provided by Liu et al. (2023a). Although some of the factors in assigning difficulty are more relevant from a reinforcement learning perspective, e.g. the sparsity of the reward, many of them also carry across to our more supervised learning setting, such as the number of valid actions and the size and richness of the vocabulary.

Overall, our best version of PASA (intent) achieves a 49% improvement versus CALM-PT (Yao et al., 2020). Our approach improves significantly on all difficult games besides DEEPHOME. The difficult games typically require more complex actions to be selected, which require more sophisticated interactions with the environment, such as knowing that the coal should be put in the furnace after it had been opened in Figure 4.1, thus demonstrating the benefit of our approach. In the outstanding case of DEEPHOME, we found this was due to the model over-predicting highly out-of-distribution actions (see Section 4.3.5). The generative model (CALM) however assigned much lower scores to out-of-distribution actions, due to their tendency towards more generic actions (Li et al., 2016a). Our approach also surpasses in most of the possible games, namely PENTARI, TEMPLE and LIBRARY, though by a smaller margin. This is to be expected, as most of the actions there consist of relatively simple actions such as navigation 'west', 'north' etc. which is likely to be well-represented in the pre-training data.

When comparing the different choices of latent variables, we find the best results come from the intent-based method, suggesting the chosen intents map well onto the text-based game structure. However, the latent approach also followed in a close second, supporting the idea that our approach could be generalised to other NLP tasks where an intent-based method might not be available. The persona-based method was significantly worse, although still better than the CALM approach. This may be because the personas learned in the ClubFloyd dataset did not map as well onto the Jericho Walkthroughs. Overall however, all our approaches were statistically significantly better than the CALM approaches on a t-test with p-value < 0.01.

We found some, though limited, benefit to fine-tuning CALM on the ClubFloyd dataset, suggesting the primary source of the benefit is from our architectural decision to disentangle the one-to-many relation, rather than the addition of the Jericho Walkthroughs dataset. This is further supported by our ablation—presented in Table 4.3—which compares PASA to the student model trained

solely on the hard labels from the datasets without access to the soft labels from the teacher. The purpose of this was to isolate the performance gains due to the posterior distillation which is a novelty of our approach. Again, both our latent and intent methods were statistically significantly better than this on a t-test with p-value < 0.01.

4.3.5 Error Analysis

We inspected the model predictions to further understand the driving factors behind the significant performance differences between different versions of PASA for some games (e.g. DEEPHOME and BALANCES). We found the persona model overly predicted the out-of-distribution actions 'say manaz' (the name of a spell) and 'spells' (which causes the game to tell the player which spells they know). Both of these are available as actions for the majority of each respective game, thereby amplifying the impact of the error. We hypothesise the reason the persona model specifically suffers from this problem may be because the latent variable it uses has more representational capacity, being a 1-of-375 categorical variable corresponding to the 375 users in the training data (versus 1-of-5 and 1-of-8 for the other two approaches). Thus, it may have overfitted more to the training data.

4.4 Conclusion

Overall, in this chapter we have addressed the problem of mode collapse in one-to-many tasks, which following the previous Chapter 3, formed the second part of addressing the broader mode mismatch problem (Section 1.1.1). In this chapter, we chose the domain of text-based games because the availability of the ClubFloyd dataset containing many different levels of player ability, combined with the Jericho Walkthroughs showing optimal player performance, provided an ideal structure for demonstrating our method's one-to-many learning capabilities.

However, these techniques are by no means limited to this domain. Therefore, in the next chapter, we consider the more generic framework of large language models, and how they are able to perform a range of language tasks without specialised training. In particular, we extend the idea of the user as a source of one-to-manyness, which has been a theme in this chapter via the latent variable (Section 4.2.5) and also in Chapter 3) via the persona profiles and show the different ways persona can be effectively integrated into existing large language models in order to control which modes of behaviour they exhibit and thus tame the one-to-many problem.

Chapter 5

One-to-Many for LLMs

5.1 Introduction

In the previous two chapters we addressed the problem of mode mismatch that often forms a part of the one-to-many problem (Section 1.1.1). A common theme across these two chapters was how the user is a key driver of one-to-manyness, due to uncertainty around their personality, preferences and objectives. While we presented two architecture specific ways to address this, in this chapter we seek a more generic way to achieve this. This is because increasingly modern machine learning is characterised by general-purpose large language models (Section 2.5), which, given they operate primarily in the domain of text, do not always have the architectural flexibility of doing things such as us adding a latent variable to them as in Chapter 4. They are often too large to perform training on consumer-grade GPUs, and so it would be better if we could achieve personalisation by leveraging their existing capabilities, rather than trying to force new behaviour into them.

As such, this chapter can be seen as the first part of addressing the problem of agnosticism of inputs and outputs introduced in Section 1.1.2. As aforementioned, our goal here is to make solutions to the one-to-many problem more flexible and task-agnostic. This is in order to make them compatible with LLMs as they continue to see improvements, making the one-to-many techniques something that can be bolted on existing LLMs.

As a test bed, we consider the setting of LLMs trained on human preference data, which is currently the most popular type of LLM, seen in interactive AI applications such as ChatGPT, Claude and Gemini. However, currently these systems are trained to reflect the *average* user's preferences, without considering inter-user variability. In many cases however, users may reasonably have completely different preferences of model behaviour for a given identical query. For instance, when asking for a summarisation of a research paper, a layperson would have entirely different requirements than an expert, in terms of level of

detail, assumptions of prior knowledge, and user's objective. Because of this, the task is inherently one-to-many, driven by uncertainty over who the user is.

A naive approach to this problem would be to fine-tune one LLM per user. However, efficiently deploying LLMs typically requires a single set of weights. There may also be insufficient data user, requiring a minimum number of samples before any meaningful improvement beyond zero-shot performance is observed. Finally, it prevents the model from learning from the relationship between different users, as the parameters per user are independent. To mitigate these issues, one may train a single model on all users' data, with the specific user being a prefix as part of the prompt. However, this incurs major privacy risks in the form of model inversion attacks, whereby user A may be able to extract user B's training data (Fredrikson et al., 2015).

As a test bed, we consider the setting of LLMs trained on human preference data, which is currently the most popular type of LLM, seen in applications such as ChatGPT, Claude and Gemini. However, currently these systems are trained to reflect the *average* user's preferences, without considering inter-user variability. In many cases however, users may reasonably have completely different preferences of model behaviour for a given identical query. For instance, when asking for a summarisation of a research paper, a layperson would have entirely different requirements than an expert, in terms of level of detail, assumptions of prior knowledge, and user's objective. Because of this, the task is inherently one-to-many, driven by uncertainty over who the user is. As a result, current approaches often focus on non-parametric solutions to injecting personalisation into the model. In particular, in-context learning is often used (Brown et al., 2020; Dong et al., 2023), whereby past interactions of the user are reformulated as part of the prompt (Geng et al., 2022; Dai et al., 2023). As the number of user interactions may scale beyond the point of scalability to include every interaction in the prompt, several studies employ retrieval-augmented generation (RAG) to retrieve a handful of relevant examples which are added to the prompt (Wo'zniak et al., 2024; Salemi et al., 2024). However, there are several challenges associated with this approach: (1) simply providing a few examples may make it difficult to properly abstract about a user's preferences, which may be multifaceted and complex; (2) increasing the number of examples may improve performance and help with the aforementioned problem, but incurs increased latency and API cost due to prompt length.

An alternate approach, to the best of our knowledge unconsidered, is to use classifier guidance to drive the behaviour of the generator. In particular, one could train a classifier to predict where a specific partially completed output is likely to have been generated by a particular user. Current approaches to achieving this typically involve directly manipulating the generator's logits, to steer them towards the target class (Yang and Klein, 2021). Compared to vanilla classifier guidance, there are unique facets to personalisation classifier guidance does not account for. In particular, adjacent work on personalised search observes that personalisation is most beneficial for high entropy queries, i.e. when there is likely to be significant inter-user disagreement (Teevan et al., 2008). Motivated

by this, we design and implement Uncertainty Weighted Classifier Guidance (**'UWCG'**). UWCG combines a classifier guidance backbone, with a novel uncertainty weighting term, which provides token-level control over the level of influence the classifier should have during the generation process. Intuitively, for high-certainty words, where there is likely to be little inter-user disagreement, this terms downweights the classifier. Conversely, when there is large amounts of uncertainty, implying high inter-user disagreement, this term allows the classifier to greater influence the generator.

Overall, the contribution of this chapter is to implement and test a novel architecture – UWCG – that can be flexibly integrated into the current set of LLMs to solve one-to-many problems that arise in modern machine learning tasks, without requiring any particular architectural changes to those LLMs.

5.2 Related Work

Prompt-based Personalisation There are broadly two approaches to personalisation in LLMs, though not necessarily mutually exclusive. Firstly, leveraging the powerful in-context learning capabilities of LLMs (Brown et al., 2020; Dong et al., 2023), retrieval-augmented generation (RAG), aims to retrieve relevant examples from a pool of previous user examples (Mazaré et al., 2018b; Wo'zniak et al., 2024). Another approach is to condition the LLM on a user persona. The structure of this persona often varies: This can be obtained via crowdsourcing, whereby humans manually write the textual persona they wish the model to emulate (Zhang et al., 2018b; Irvine et al., 2023). Further work seeks to automate this through using LLMs to generate a summary based on a list of user examples (Richardson et al., 2023). These works may also implement a hybrid prompt that concatenates both retrieved examples and user summaries (Li et al., 2023).

Parametric Personalisation An alternative approach to personalisation is to update the model's parameters to reflect new user data. Several works explore improving the retriever to better calibrate it to retrieve relevant user data, such as through reinforcement learning and knowledge distillation (Salemi et al., 2024; Mysore et al., 2023). Other works explore directly fine-tuning the LLM on user specific data (Wo'zniak et al., 2024).

Controlled text generation represents a broad band of techniques in which an unconditional generation model is biased towards generations of a particular class. Early work learned domain-specific control 'codes' Keskar et al. (2019). More recent work has shifted towards classifier guidance, enabling more decoupling between the unconditional and class-conditional components: here, a separate classifier guides token-by-token generation (Krause et al., 2020; Yang and Klein, 2021; Shuster et al., 2021b; Arora et al., 2022). Alternatively, other work

has explored classifier-free guidance, in which both a conditional and unconditional language model perform parallel decoding with merged logits (Sanchez et al., 2023). So far as we are aware, our work is the first to explore the application of classifier guidance to personalised text generation and classification.

5.3 Method

In this section we introduce our method. Section 5.3.1 introduces key notation and overviews the personalisation task. Section 5.3.2 summarises the RAG framework that is used both in our method and baselines. Section 5.3.3 introduces the autoregressive generator used as a backbone for all methods tested. Section 5.3.4 describes how we apply the classifier guidance framework to the task of personalisation. Section 5.3.5 describes our novel uncertainty weighting term, that helps the classifier to provide additional guidance on terms where there is most likely to be inter-user disagreement.

Algorithm 1 Uncertainty-weighted Classifier Guidance

Input Generator Θ, Classifier Φ, prompt x, query q, number of candidates M.
Output prediction $\hat{y}$

procedure UWCG(Θ, Φ, x, q, M)
  for $t \leftarrow 1$ to T **do**
    $p_\Theta(y_t|x, y_{<t}) \leftarrow \Theta(x, \{\hat{y}\}_{t=1}^{t-1})$
    $topM \leftarrow$ *M*-argmax$(p_\Theta(y_t|x, y_{<t}))$
    $bestScore, bestToken \leftarrow 0.0, None$
    for $m \leftarrow 1$ to m **do**
      $\hat{y}_{t,m} \leftarrow topM[m]$
      $p_\Phi(u|x, y_{\leq t}) \leftarrow \Phi(y, \{\hat{y}\}_{t=1}^{t})$
      $p(\hat{y}_{t,m}|x, y_{<t}, u) \leftarrow p_\Theta(y_t|x, y_{<t}) \cdot p_\Phi(u|x, y_{\leq t})$
      if $p(\hat{y}_{t,m}|x, y_{<t}, u) > bestScore$ **then**
        $bestScore, bestToken \leftarrow p(\hat{y}_{t,m}|x, y_{<t}, u), \hat{y}_m$
      end if
    end for
    $y_{\leq t} \leftarrow y_{<t} + bestToken$
  end for
  return y
end procedure

5.3.1 Task definition

Suppose we have a dataset of tuples $\mathcal{D} = \langle x, Z, u, y \rangle_i^I$. x is the sample-specific query. $\{z_k\}_{k=1}^N$ is the personalised set of N documents that are specific to a user u. y is the ground-truth label according to u, comprising tokens $\{y_1, ..., y_T\}$. We construct the document set Z by taking all $\langle$ query, label $\rangle$ pairs from the training set for u.

Our goal is to learn a function $\Phi(x, Z)$ to predict $\hat{y} = \{\hat{y}_1, ...\hat{y}_T\}$ [1] which should be as close as possible to y, according to some arbitrary metric (e.g. accuracy, F1). Following previous work on personalisation (Salemi et al., 2024), we assume that the underlying LLM's parameters remain frozen due to the potentially large cost associated with fine-tuning and privacy concerns around user training data regurgitation from model inversion attacks (Fredrikson et al., 2015). As such, during inference the LLM is also provided with a task-specific prompt p, which is pre-pended to the query to enable zero-shot capabilities (Brown et al., 2020).

Despite keeping the LLM fixed, however, various strategies remain open to enable personalisation, such as through optimising a retriever to inject relevant user passages into the prompt or guiding the decoding process through a specialised classifier.

5.3.2 Retrieval-augmented Generation

In this section we briefly describe Retrieval-augmented Generation (RAG), a popular technique for controlling model behaviour. First, the set documents $z \in Z$ is encoded into vectors using some arbitrary embedding model Ψ to obtain embeddings $\{\mathbf{z}_1, ..., \mathbf{z}_N\}$. Then the user query q, which is a subset of the prompt x, is also encoded to obtain $\mathbf{z}_q$. In our case, we simply use the sequence to be classified or the current dialogue context, removing in both cases the fixed system prompt. Then, we select the most similar documents according to cosine similarity:

$$\underset{1}{\overset{N}{k\text{-argmax}}} = \frac{\mathbf{z}_q}{|\mathbf{z}_q|} \cdot \frac{\mathbf{z}_n}{|\mathbf{z}_n|} \tag{5.1}$$

5.3.3 Generator

The backbone of our approach is a large language model. Given a prompt x, this aims to generate an output through autoregressively predicting tokens, where the final output probability can be factorised as:

$$p_\Theta(y|x) = \prod_t^T p_\Theta(y_t|x, y_{<t}) \tag{5.2}$$

For classification tasks, this involves generating the textual labels for the classes themselves.

[1] Note for simplicity of notation we use the same T to represent the sequence length of the ground-truth and of the prediction although these may differ in practice.

5.3.4 Classifier Guidance

Classifier guidance aims to transform an unconditional generation model $p_\Theta(x)$, into a conditional generation model $p(x|y)$, through the intermediation of a classifier $p_\Phi(y|x)$. This is achieved via the following Bayesian decomposition (Yang and Klein, 2021):

$$p(y_t|x, y_{<t}, u) \propto p_\Theta(y_t|x, y_{<t}) \cdot p_\Phi(u|x, y_{\leq t}) \tag{5.3}$$

Our classifier comprises a standard bidirectional transformer (Devlin et al., 2019b), with a linear layer which is connected to the output of the `[CLS]` token which is pre-pended to the input sequence. The input sequence comprises both the query q and the partially completed generation of y. Specifically, in order to train the classifier to perform classification at any step of the generation process, we preprocess each dataset by randomly splitting the completion y tokenwise into a prefix $p = \{p_1, ..., p_t\}$ and a suffix $s = \{s_{t+1}, ..., s_T\}$. We omit the suffix, and train the model to predict the user, given the concatenation of the query and prefix, using crossentropy loss:

$$\mathcal{L}_{CE} = \mathrm{CLS}(\mathrm{MLP}(\Phi([q; p])[0]), u) \tag{5.4}$$

During inference, during each timestep t of the total sequence generation length T, we select the top M tokens according to the generator Θ. For each token, we concatenate this to to the query and sequence of previously generated tokens: $\{q, \hat{y}_1, ..., \hat{y}_t\}$. We apply each such sequence to the classifier, to obtain a score for each possible next token and merge these scores with the generator scores using Equation 5.3.

5.3.5 Uncertainty Weighting

Intuitively, previous work on personalisation in adjacent fields such as search have found personalisation to be most effective on more uncertain queries (Teevan et al., 2008). We therefore model this through a sequence of uncertainty terms $B = \{\beta_1, ..., \beta_T\}$, where β_t is defined as:

$$\beta_t = 1 + H[p_\Theta(y_t|x, y_{<t})] \tag{5.5}$$

and $H(\cdot)$ is the entropy function:

$$-p_\Theta(y_t|x, y_{<t}) \cdot \log p_\Theta(y_t|x, y_{<t}) \tag{5.6}$$

Note, that as $\min_{\Theta \in \mathcal{P}} H(p_\Theta(y_t|x, y_{<t}) = 1$, where $\mathcal{P}$ is the parameter space, the initial 1 constant is used to prevent reverse personalisation, i.e. the minimum possible personalisation becomes no personalisation. Thus, our overall classifier

Metric \ Dataset	Classification		Generation
	GoEmotions	Unhealthy Conversations	Friends
Train	156k	36k	51k
Val	19k	4k	6k
Test	19k	4k	6k
# Users	82	498	696
# Examples per User	1896	71	74

Table 5.1: Statistics of the datasets used. 'User' here refers to the annotator for the classification datasets and the speaker for the Friends dataset. Examples for user are taken from the Train set for each respective dataset.

guidance equation can be factorised as:

$$p(y|x,u) \propto \prod_{t}^{T} p_{\Theta}(y_t|x, y_{<t}) \cdot \beta_t \cdot p_{\Phi}(u|x, y_{\leq t}) \tag{5.7}$$

5.4 Experiment

In our experiments, we aim to answer the following research questions:

- **RQ1:** Does personalised retrieval offer any benefit versus non-personalised retrieval?
- **RQ2:** What is the effect of varying the number of retrieved examples?
- **RQ3:** How does classifier-guidance compare to RAG in performance?
- **RQ4:** What is the effect of combining classifier-guidance with RAG?

5.4.1 Experimental Setup

As our LLM base, we use the `Flan-T5-Large` [2] and `Flan-T5-Base` [3] (Chung et al., 2022) models, which are both instruction-tuned encoder-decoder transformers with 780M and 240M parameters respectively. We chose two different sized models in order to further test how consistently many of the trends would hold up. For performing RAG, we use the `MiniLM-L6-v2` retriever [4]. Examples are taken from the entire training set, with each example comprising the concatenation of both (i) the text to be classified / dialogue context and (ii) the labels /

[2] https://huggingface.co/google/flan-t5-large
[3] https://huggingface.co/google/flan-t5-base
[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

		Classification				Generation	
		GoEmotions		Unhealthy Conversations		Friends	
Model	Method	Acc	F1	Acc	F1	RGE-1	RGE-L
Flan-T5 Large	ZS	20.1	20.8	2.5	5.8	8.7	8.2
	RAG-1	21.0	20.8	8.7	7.7	9.1	8.7
	RAG-3	21.0	20.8	8.7	7.7	9.4	9.0
	P-RAG-1	21.4	20.8	6.6	7.4	9.5	9.0
	P-RAG-3	22.8	21.0	10.6	8.2	9.4	9.3
	UWCG-ZS	24.4	23.7	5.6	8.0	8.9	8.4
	UWCG-1	26.2	23.8	16.1	10.8	9.6	9.2
	UWCG-3	**26.9**	**23.9**	**23.9**	**12.6**	9.4	9.0
Flan-T5 Base	ZS	14.1	16.5	1.6	5.2	7.7	7.6
	RAG-1	16.2	17.4	3.6	5.8	7.9	7.7
	RAG-3	15.2	16.3	0.2	3.8	6.7	6.6
	P-RAG-1	16.7	17.7	2.4	5.4	8.4	8.2
	P-RAG-3	14.9	16.2	0.5	4.6	8.4	8.3
	UWCG-ZS	16.3	18.7	2.6	6.5	7.9	7.6
	UWCG-1	**19.5**	**19.9**	**3.8**	**7.0**	8.7	8.4
	UWCG-3	17.4	18.6	0.4	5.6	8.7	8.4

Table 5.2: Evaluation results on the test sets for personalised classification and generation tasks, comparing the effects of baseline approaches with and without our method UWCG. **Bold** indicates best result, underline indicates second best result.

reply. We pre-compute their L2-normalised vector representations and retrieved using cosine similarity with the query, which consists of solely component (i). As previously indicated, the parameters of the models remain frozen throughout the evaluation process.

We instantiate the classifier using the `gte-base model` (Zhang et al., 2024) [5], a 137M parameter BERT-like encoder, which has achieved state-of-the-art performance within its size category on the MTEB leaderboard (Muennighoff et al., 2022). We train using HuggingFace's `transformers` library with the AdamW optimiser (Loshchilov and Hutter, 2019) with a learning rate of 2e-5 for up to 10 epochs until convergence with a batch size of 16 using an Nvidia RTX 3060 Ti. All other hyperparameters are kept unchanged.

5.4.2 Datasets

We evaluate our method across two classification and one generation dataset. Table 5.1 summarises the dataset statistics. **GoEmotions** (Demszky et al., 2020) comprises emotion annotations on 58k Reddit comments from 82 annotators. Each comment has multiple annotations from different annotators, resulting in a total of 212k samples. **Unhealthy Conversations** (Price et al., 2020) comprises 230k annotations from 588 annotators of 44k news comments. Comments are classified according to in which way the comment is deemed unhealthy, i.e. hostile, dismissive, etc. For both tasks, the aim is to predict the label, given the

[5] https://huggingface.co/Alibaba-NLP/gte-base-en-v1.5

comment and annotator id. **Friends** (Chen and Choi, 2016) comprises the TV scripts from the popular US sitcom by the same name. Given the current conversation context and speaker, the task is to predict the next utterance. As friends is a well known TV series with many details about its characters likely present in the LLM's pre-training data we replace the speaker prefixes in the dialogue with anonymised ids. This puts the dialogue in a more comparable situation to the classifier datasets, which both feature anonymised annotators.

5.4.3 Baselines

As previous works on personalisation of LLMs have largely focused RAG-based techniques to modify the prompt, our method, which focuses on modifying the output logits is compatible with these approaches. We therefore report performance for the following baselines with and without our method:

- 0-shot: Non-personalised version of the model, given only the prompt without any user specific example.
- 1-shot: Selects the top-1 scoring user example, given the current query.
- 3-shot: Selects the top-3 scoring user examples, given the current query.

Further, in order to analyse the effects of personalisation in particular, for the baselines we consider two different settings, RAG, in which examples are retrieved from the entire training set, and P-RAG in which examples are retrieved solely from the relevant user. We note that the default RAG setting in some cases could lead to privacy issues in a deployment setting as, being part of the prompt, another user may be able to adversarially extract it (Zhang and Ippolito, 2023). However, we include it here for the sake of empirical study and in any case, we show in Section 5.4.8 how limiting the retrieval to same-user only examples typically improves performance.

5.4.4 Metrics

For the classification tasks, we evaluate using both absolute accuracy and macro F1 score. For generation tasks, we evaluate using ROUGE-1 and ROUGE-L which measure single term overlap and longest subsequence overlap respectively (Lin, 2004). Note that techniques such as human evaluation are impractical to implement in the personalised setting, as the quality of a response is dependent on the specific persona. Future work may explore evaluating on live systems, where real users are available.

5.4.5 Main results

Table 5.2 presents our main results. We analyse the effect of applying classifier guidance to both the zero-shot, one-shot and few-shot ($K = 3$) setting. Across all datasets and settings we find consistent improvement. Notably, we find that this benefit is orthogonal to the benefits of personalised RAG, as the performance delta between versions without and with our method remains consistent regardless of K-shot setting. This is important as it indicates that our method can be used as an enhancement to traditional RAG setups, rather than necessarily as a replacement.

Across datasets we find the largest improvement in Unhealthy Conversations. As this is by far the hardest of the three tasks due to class boundaries being often ambiguous (i.e. the difference between *antagonistic* and *hostile*), we hypothesise that particular annotators may have personal preferences around where to draw these boundaries, hence facilitating great benefit from classifier guidance. By contrast the weakest benefit was found in the Friends dataset. Although we did attempt to anonymise the dialogue, we hypothesise that it may still be possible to contextually infer who the current speaker is, in which case the non-guided models can still benefit from the examples implicitly available from eariler dialogue turns, plus any prior knowledge about the character if an exact match can be made. Further research may test this through considering other dialogue datasets created by crowdworkers or limiting it to a single-turn setting.

5.4.6 Varying number of retrieved examples

Previous work on in-context learning observes general improvement in performance as the number of examples increases (Agarwal et al., 2024). As the number of retrievable examples per user is only a fraction of the overall knowledge base in the case of vanilla RAG, it was initially non-obvious whether increasing K would simply lead to the retriever injecting less relevant distracting examples into the context window.

In Figure 5.1, we evaluate this for Flan-T5 by considering varying numbers of examples, including up until the 512 token window is exhausted. This involves up to 7-shot for GoEmotions, 10-shot for Unhealthy Conversations. We omit the Friends dataset, as the context window is typically maxed out within 3-shot and therefore results are already present in the main results. In all cases, for the large model, we find consistent improvements across all metrics. We further observe these improvement gains stack with classifier guidance. For the base model however, we find a decline in performance above 1-shot. We hypothesise this may be that smaller less powerful models struggle to simultaneously attend to both the prompt and the given examples. In both models however, the largest benefit occurs when going from 0-shot to 1-shot, indicating providing a single example provides powerful disambiguating information on how the answer should be

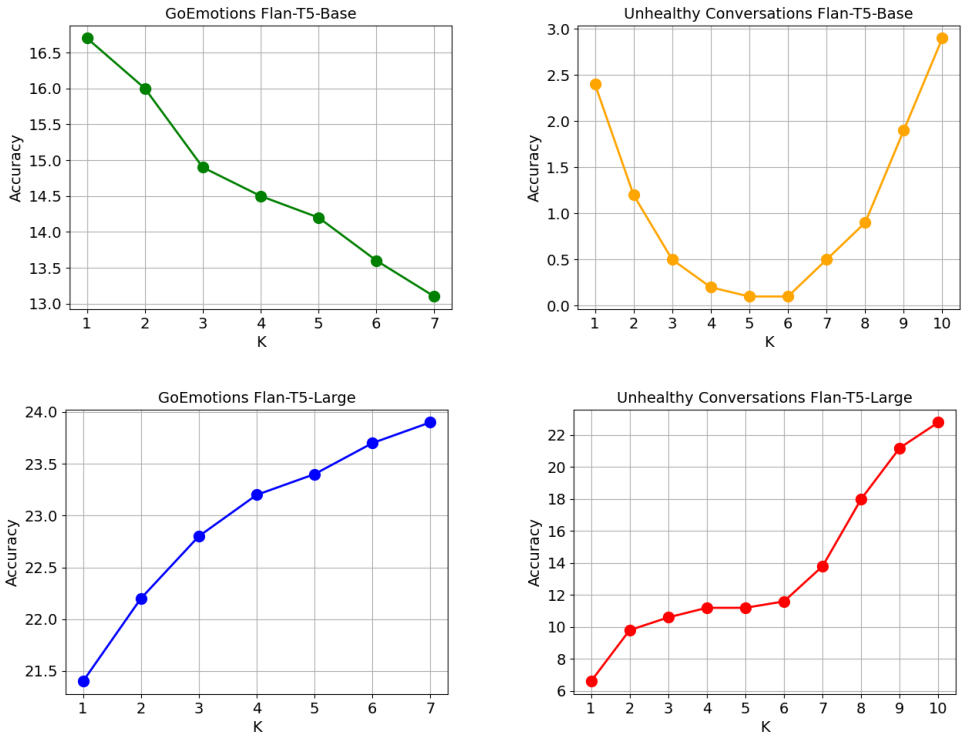


Figure 5.1: Effect of different numbers of retrieved examples (K) for the GoEmotions and Unhealthy Conversations datasets under different model sizes.

structured, where a text-based prompt alone may otherwise be ambiguous.

5.4.7 Importance of retriever

As many approaches to RAG rely on the assumption that semantic vector similarity between query and example is a good proxy for end task relevance, we test the effects of removing the neural retriever and simply randomly selecting examples from the specific user. We found that while there is still a benefit, albeit slightly weaker, in the 1-shot case, the benefits beyond this are more dataset dependent, indicating too many random examples may simply add distracting noise to the prompt.

As examples are retrieved on the basis of cosine similarity, in Table 5.4 we perform an analysis of the retriever baseline approaches for $K = 1$. In particular, we assess how the accuracy and ROUGE-1 score for the classification and generation tasks respectively are correlated with the cosine similarity of the retrieved examples. We find that absolute cosine score is not predictive of increased accuracy for the classification tasks and is only weakly predictive in the generation case. Further, while the personalised models consistently outperform non-personlised ones, the non-personlised ones also show a higher cosine similarity on average. This suggests that absolute cosine similarity is not necessarily predictive of the usefulness of an example. We hypothesise that because the retriever

	GoEmotions		Unhealthy Conversations	
	Acc	F1	Acc	F1
UWCG				
$K = 0$	**24.4**	**23.7**	**5.6**	**8.0**
$K = 1$	**26.2**	**23.8**	**16.1**	**10.8**
$K = 3$	**26.9**	**23.9**	**23.9**	**12.6**
Vanilla CG				
$K = 0$	23.1	23.1	2.8	6.6
$K = 1$	25.3	23.3	8.2	8.4
$K = 3$	26.8	23.4	13.6	10.0

Table 5.3: Ablation study on the effect of using our method, UWCG, versus vanilla classifier guidance using personalised retrieval with the **Flan-T5-Large** model. **Bold** indicates superior result for respective number of K.

Method \ Dataset	Classification				Generation	
	GoEmotions		Unhealthy Conversations		Friends	
	Cosim	Pearson	Cosim	Pearson	Cosim	Pearson
RAG-1	0.483 (0.0805)	-0.018	0.527 (0.094)	-0.084	0.735 (0.0804)	0.189
P-RAG-1	0.393 (0.082)	0.14	0.344 (0.0984)	-0.085	0.701 (0.0954)	0.211

Table 5.4: Analysis of mean (std.) of the cosine similarity of the example retrieved by RAG as well as the pearson correlation between cosine similarity and accuracy / ROUGE-1.

is out-of-the-box, it may not be representing key aspects about the user well in its embedding, but may just be retrieving examples that are similar to the query.

5.4.8 Personalised versus non-personalised RAG

As RAG is known to provide benefits in the vanilla non-personalised setup, we further try to quantify how much of the benefit can be allocated to the RAG alone, which may provide structural task information, and, in the case of classification, elucidate broader class distributional information – and how much can be assigned to the personalision component, which contains user preference information. We find there is consistent benefit to personalisation, although the exact split varies by dataset, with roughly 75% of the improvement coming from RAG and the remaining 25% coming from personalisation.

5.4.9 Ablation Study

In Table 5.3 we perform an ablation to analyse the performance of our bespoke classifier guidance system, UWCG, compared to a vanilla classifier guidance system. In particular, we explore the effect of removing the uncertainty weighting term (i.e. Equation 5.3) under different parameters of K. We find that performance consistently drops when this feature is removed across all values of K. This is especially noticeable in the Unhealthy Conversations dataset, suggesting it is particularly important for more challenging datasets, where the generator's uncertainty is likely to be higher.

5.5 Conclusion

In this chapter, we have addressed the first part of the problem of making solutions to the one-to-many problem more input agnostic (Section 1.1.2). In contrast to the previous two chapters, which relied on more task-specific, bespoke solutions, we demonstrated a task-agnostic means of achieving this, UWCG, a classifier guidance approach to personalisation, that introduces a novel uncertainty weighting term to allow variable levels of guidance depending on the level of personalisation required.

While so far we have demonstrated the applicability of these techniques to text, in the next chapter we explore how these techniques can be applied to modalities beyond text, in particular medical image segmentation, supporting the second part of the problem introduced in Section 1.1.2.

Chapter 6

Different Modalities

6.1 Introduction

In the previous chapter, we tackled the first part of the problem around making the techniques for the one-to-many problem as agnostic as possible (Section 1.1.2). However, the limitation was still that we were only working in the domain of text, as personalisation was achieved in part by filling the context of the model with historical user information, a limitation shared with the chapters that came before (3, 4). However, in order to achieve truer agnosticism to the types of inputs and outputs our model can deal with–and, to make the one-to-many techniques presented here useful in the future to a broader audience–we seek in this chapter to show how they can be applied to other modalities. In particular, we choose medical image segmentation. We choose this as it is a natural fit for the interactive one-to-many setting (Section 2.1), due to uncertainty from the low-contrast setting and inter-annotator disagreement.

Following in the spirit of Chapter 5, we again opt to use a pre-trained foundation model, as these are increasingly becoming the *de facto* approach across numerous fields in machine learning (Bommasani et al., 2021; Brown et al., 2020; Rombach et al., 2021) and so this supports the objective of agnosticism presented in Section 1.1.2. In image segmentation, the recently released Segment Anything Model (SAM) (Kirillov et al., 2023) has demonstrated state-of-the-art semi-automatic zero-shot capabilities, through re-framing the prompt as interaction information from the user such as clicks, bounding boxes or masks, which guide the annotation of the region of interest (Santos et al., 2022) (Section 2.6.

There is growing interest in applying SAM to Medical Image Segmentation (Ma et al., 2023; de Oliveira et al., 2023; Roy et al., 2023; Mohapatra et al., 2023). Previous approaches to medical imaging focused on supervised training of bespoke models for each task, requiring numerous manually labelled examples (Roy et al., 2023). Yet, this presents several limitations: the requirement for a trained clinician renders annotation costs extremely high (Rahimi et al., 2021);

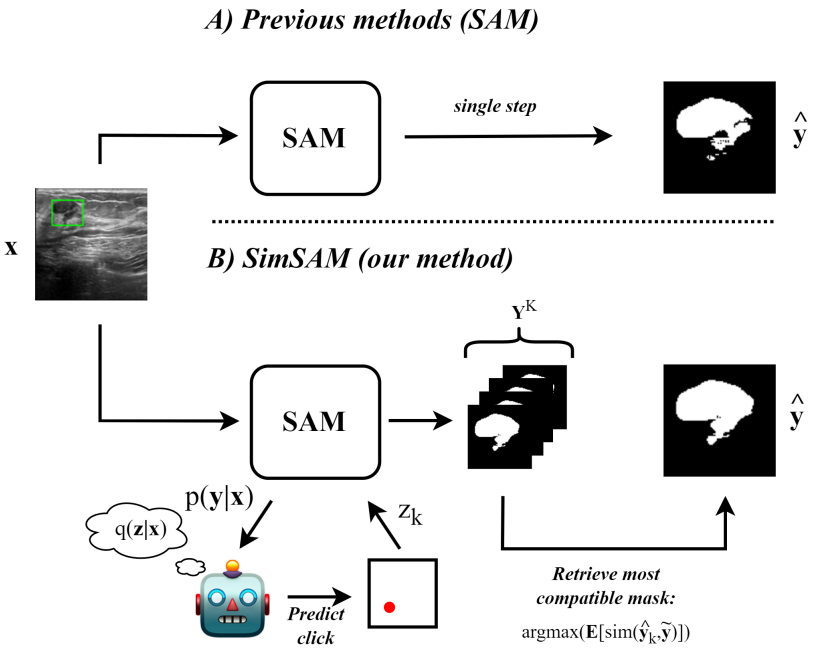


Figure 6.1: (A) Previous methods predict annotation mask in a single step; (B) SimSAM, our method, simulates possible click locations to provide an additional prompt to the model and retrieves the most compatible mask from a pool of generated masks. Example is from Breast Ultrasound Scan test set (Al-Dhabyani et al., 2019).

further, privacy restrictions may prevent sharing of patient data (Adnan et al., 2021), limiting the availability of the kind of large-scale datasets seen in other domains such as NLP; finally, these models often show poor generalisability out of the lab, e.g. due to variability in modality or device used to obtain the images (Ali et al., 2022). Resultantly, SAM could greatly speed up existing clinical pipelines, by enabling rapid semi-automatic segmentation of medical images (Roy et al., 2023).

However, unlike natural images, medical images often have significant uncertainty around their contours, e.g. due to unclear boundaries between healthy and unhealthy tissue, or noise from low-contrast media. This leads to substantial expert disagreement about an image's correct annotation (Kohl et al., 2018). Appropriately determining these contours has major downstream consequences, such as deciding how invasive a surgery will be in the case of a tumour (Monteiro et al., 2020). When attempting this as a zero-shot task, SAM is known to perform particularly poorly at appropriately segmenting the edges of the image (Roy et al., 2023), resulting in significant additional manual corrections being required by the clinician to obtain satisfactory accuracy.

We observe that unlike segmentation models which produce a dense pixel-wise output in a single forward pass, a human clinician creates annotations in more of a sequential process, enabling more careful consideration between alternate hypotheses for the contours. Concurrently, emergent capabilities have been dis-

covered in many foundation models through chain-of-thought prompting (Wei et al., 2022) – i.e. feeding the model's outputs back into itself as a prompt, to enhance its final generation. This enables the model to draw upon 'dark knowledge' (Hinton et al., 2015b) – information learned during training, but not immediately visible in its outputs – to guide its predictions. Intuitively, this allows a model to break a problem down into incremental steps, rather than requiring it to make an immediate prediction. So far however, there has been a lack of work exploring how this principle can be applied to interactive segmentation models, whose 'prompts' are clicks and bounding boxes, rather than text.

In this chapter, we introduce **Sim**ulated Interaction for **S**egment **A**nything **M**odel (**SimSAM**), a zero-shot extension to SAM that significantly enhances SAM's out-of-the-box performance, without requiring any additional training. Specifically, SimSAM uses a carefully-designed click simulation mechanism that extracts knowledge about locations of user clicks to form additional prompts to enhance its predictions. We further propose a method for aggregating these predictions, which maximises the compatibility of the outputted images. Figure 6.1 demonstrates how in contrast to vanilla SAM which predicts the mask in a single step, SimSAM is able to iteratively improve its prediction through producing multiple masks from self-generated prompts, and aggregating over these masks. We evaluate our method across three publicly available medical imaging datasets. Quantitatively, we find that our approach consistently outperforms SAM across all three datasets, with up to 15.5% improvement in contour segmentation accuracy. Qualitatively, we demonstrate the superior annotations of our approach, and show how our approach is able to generate more robust masks that mitigate many of the pitfalls of SAM.

6.2 Methodology

Figure 6.1B overviews our method. Given an input image of N pixels $\mathbf{x} = \{x_n\}_{n=1}^{N}$, our goal is to predict a binary mask $\hat{\mathbf{y}}$, that matches an expert annotation $\mathbf{y}$, using the Segment Anything Model (SAM). We first show how $\hat{\mathbf{y}}$ can be predicted by marginalising over a known distribution of user clicks, then show how this process can be approximated by SAM.

6.2.1 Segment Anything Model

SAM is a foundation segmentation model trained on over 1B masks from 11M images (Kirillov et al., 2023), with the ViT backbone transformer encoder (Dosovitskiy et al., 2020). After encoding an image, the model enables an output mask to be iteratively refined through conditioning on various 'prompts', e.g. user clicks and bounding boxes, which are attended to by the model's decoder (see Section 2.6).

6.2.2 Marginalising over Prompts

We assume that each mask $\mathbf{y}$ is conditioned not only on the input image $\mathbf{x}$ but also on a prompt z whose possible values represent the available points a user might click $\{z_n\}_{n=1}^{N}$. The ground-truth probability distribution over user clicks is then given by $p(z|\mathbf{x})$. Given input image $\mathbf{x}$, SAM first produces a dense pixel-wise probability distribution $p(\mathbf{y}|\mathbf{x})$. Then, we sample a click from $p(z|\mathbf{x})$ and make a new prediction $\hat{\mathbf{y}}$ conditioned on this click. Thus, we estimate the probability for the output mask, by marginalising over user clicks as follows:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{n}^{N} p(\mathbf{y}|\mathbf{x}, z_n)p(z_n|\mathbf{x}) \tag{6.1}$$

6.2.3 Simulating User Clicks

We do not in practice have access to a ground-truth distribution $p(z|\mathbf{x})$, and requiring a user to provide this would defeat the purpose of improving SAM's performance without requiring any additional manual human annotation. Instead, we would like to obtain some distribution $q(z|\mathbf{x})$ that approximates this. Let $\mathbf{e} = \mathbf{fp} \cup \mathbf{fn}$ be the error mask of incorrectly annotated pixels, comprising the union of false positives $\mathbf{fp}$ and false negatives $\mathbf{fn}$. We follow the assumption from previous work that the user will click one of these pixels (Liu et al., 2022). Then, although the model is not explicitly trained to predict an error mask, we observe that the model can implicitly provide a zero-shot approximation of the probabilities for these, simply by transforming the original probability mask $p(\mathbf{y}|\mathbf{x})$, namely:

$$p(e_n = 1) = 1 - \text{abs}[p(y_n|\mathbf{x}) - 1] \tag{6.2}$$

We emphasise that this click simulation is inferred entirely from the zero-shot probabilities of SAM, *without* requiring any additional gradient updates to the model.

6.2.4 Top K Approximation

Even approximating $p(z|\mathbf{x})$ with $q(z|\mathbf{x})$, it is intractable to calculate Equation (6.1) exactly, as the number of possible clicks is equal to the number of pixels in the image N. We therefore approximate this through taking the top K clicks:

$$p(\mathbf{y}|\mathbf{x}) \approx \frac{1}{K} \sum_{k}^{K} p(\mathbf{y}|\mathbf{x}, z_k), z_k \in \text{TopK}[p(z|\mathbf{x})] \tag{6.3}$$

6.2.5 Image-level Aggregation

While the above method enables marginalisation over pixel-level outputs, it does not explicitly consider the interdependencies between pixels. Simply independently averaging pixel-values across each of the masks may fail to produce a mask that is overall coherent. We therefore limit our final output $\hat{\mathbf{y}}$ to retrieving from the space of masks generated by SAM: $\mathbf{Y}^K = \{\hat{\mathbf{y}}_k\}_{k=1}^K$. To select the mask that is most representative of the set of generated masks, we consider the compatibility of each mask in the set to every other mask. Concretely, we instantiate this by selecting the mask that has the highest overall image-level similarity to the other masks:

$$\hat{\mathbf{y}} = \underset{k=1:K}{\operatorname{argmax}} \, \mathbb{E}_{\tilde{\mathbf{y}} \sim p(\mathbf{y}|\mathbf{x})}[\operatorname{sim}(\hat{\mathbf{y}}_k, \tilde{\mathbf{y}})] \tag{6.4}$$

where for simplicity we use intersection-over-union (IoU) to represent our similarity function $\operatorname{sim}(\cdot, \cdot)$ (Monteiro et al., 2020):

$$\mathbb{E}_{\tilde{\mathbf{y}} \sim p(\mathbf{y}|\mathbf{x})}[\operatorname{sim}(\hat{\mathbf{y}}_k, \tilde{\mathbf{y}})] \approx \frac{1}{K} \sum_{k'}^{K} \mathrm{IoU}(\hat{\mathbf{y}}_k, \tilde{\mathbf{y}}_{k'}) \tag{6.5}$$

6.3 Results

6.3.1 Experimental Setup

We compare our method principally to the out-of-the-box SAM model (Kirillov et al., 2023), which previous studies use as a SoTA zero-shot model (Roy et al., 2023; de Oliveira et al., 2023), i.e. given input image $\mathbf{x}$, the model predicts $\hat{\mathbf{y}}$ without conditioning on any clicks. We also provide an indication of SAM's upper bound, compared to zero-shot performance, by fine-tuning on the ground-truth annotations for each dataset. We freeze the encoder and update only the decoder's parameters as per Ma et al. (2023) and optimise the model using DICE loss (Section 2.3). Note, for fair comparison we do not compare with MedSAM (Ma et al., 2023) as their approach is trained on vast amounts of medical data, and also cannot condition its predictions on user clicks, making it unsuited to our semi-automatic setting.

While problems such as low-contrast are a common feature across multiple medical imaging datasets, leading to the previously described one-to-many nature of them, they are by no means homogeneous. They vary in terms of contrast based on their modality, and the shape and size of the object to be segmented also can change the difficulty of the task. We therefore select three different datasets that capture a range of segmentation difficulties to evaluate our approach. (1) **Breast Ultrasound Scan Images ('BUSI')** (Al-Dhabyani et al., 2019) contains 437 images of breast cancer from ultrasound scans (after excluding blank images). (2) **CVC-ClinicDB ('CVC')** (Bernal et al., 2015) contains 612 images from 31

	Breast Ultrasound Scan		CVC-ClinicDB		ISIC-2016	
Method	**DSC ↑**	**NSD ↑**	**DSC ↑**	**NSD ↑**	**DSC ↑**	**NSD ↑**
(A) Fine-tuning						
SAM-FT	88.4 ± 4.3	52.6 ± 21.8	94.0 ± 5.7	81.6 ± 15.3	94.5 ± 2.8	56.8 ± 22.2
(B) Zero-shot SAM	79.3 ± 14.8	38.0 ± 21.7	86.8 ± 19.0	64.4 ± 23.3	$\mathbf{81.9 \pm 13.1}$	17.4 ± 17.1
SimSAM (ours)	$\mathbf{81.3 \pm 15.2}$‡	$\mathbf{41.6 \pm 21.8}$‡	$\mathbf{87.3 \pm 20.0}$	$\mathbf{69.2 \pm 23.4}$‡	81.8 ± 13.5	$\mathbf{19.0 \pm 18.8}$‡
(C) Ablations						
Random $q(\mathbf{z}\|\mathbf{x})$	83.6 ± 8.7	42.6 ± 23.8	82.3 ± 22.8	62.0 ± 29.8	69.4 ± 27.1	21.0 ± 18.5
Pixel aggregation	77.5 ± 18.6	39.1 ± 22.5	82.2 ± 26.9	65.8 ± 28.3	78.3 ± 18.5	17.2 ± 18.3
$K = 1$	75.8 ± 22.2	40.0 ± 23.4	80.4 ± 27.8	60.4 ± 28.8	73.4 ± 24.6	16.9 ± 18.6

Table 6.1: Mean $\pm$ standard deviation of results on the Breast Ultrasound Scan, CVC-ClinicDB and ISIC-2016 test sets, showing: (A) upper bound obtained from fine-tuning; (B) zero-shot results for our method and the baseline; (C) ablations of key components from our method. **Bold** indicates best zero-shot result. ‡ indicates statistically significant difference between SimSAM and SAM using Wilcoxon Signed Rank Test ($p < 0.01$).

colonoscopy sequences for polyps identification. (3) **ISIC-2016 ('ISIC')** (Gutman et al., 2016) contains 1279 lesion segmentations from dermoscopic images for identifying melanoma, a lethal form of skin cancer. As we focus on the semi-automatic setting, we provide a bounding box for each image, using the extremity points of the ground-truth mask to mimic an initial user input. Although our method and main point of comparison are both zero-shot and therefore require no training, to enable fair comparison to the fine-tuned version of SAM, we split the dataset 80 / 10 / 10 into training, validation and test sets respectively, except for ISIC where we use the already split test set and extract 10% of the train set for validation.

We run our models using the 94M parameter Segment Anything Model (SAM) (Kirillov et al., 2023), although our method is extensible to future pre-trained segmentation models with similar click-based interaction capabilities. We chose $K = 50$ for top K approximation, which provided a good trade-off between latency and performance.

Following previous work in medical imaging (Ma et al., 2023), we evaluate using both Dice Similarity Coefficient (DSC) and Normalised Surface Distance (NSD) to calculate the accuracy of predictions (Section 2.10). DSC is a region-based metric that computes the pixel-level harmonic mean between the ground-truth and predicted mask, while NSD is a contour-based metric that computes the consensus between the boundaries of two masks.

6.3.2 Main Results

Table 6.1B compares the zero-shot performance of SimSAM to the baseline SAM method. We find consistent outperformance across all three datasets. In terms of DSC, SimSAM obtained comparable or superior performance, showing that even though our approach is originally motivated to improve accuracy

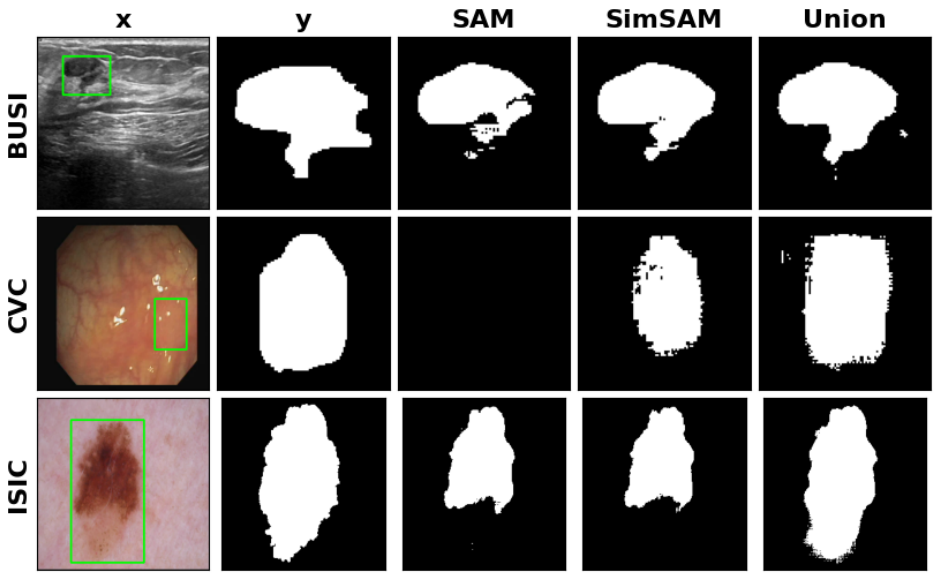


Figure 6.2: Qualitative results from the Breast Ultrasound Scan, CVC-ClinicDB and ISIC-2016 test sets. (Col. 1) Image $\mathbf{x}$ with bounding box prompt; (Col. 2) binary ground truth mask $\mathbf{y}$; (Col. 3) baseline SAM model (Kirillov et al., 2023); (Col. 4) SimSAM; (Col. 5) Union of all $K = 50$ masks generated by SimSAM.

around the contours of the image, it does not sacrifice accuracy in regional measures. In terms of NSD, SimSAM obtained consistently significant improvement measured by Wilcoxon Signed Rank Test ($p < 0.01$). This demonstrates our approach was consistently effective at improving accuracy around the region of interest's contours. We found the largest performance gains in the BUSI dataset, which is the more challenging dataset that contains significant uncertainty over edge boundaries. Table 6.1A provides an indicative upper bound by demonstrating the performance of a fine-tuned version of SAM on each dataset. In terms of latency per sample, SimSAM is only moderately slower at 397ms compared to 245ms for the baseline.

6.3.3 Ablation Study

Table 6.1C further shows the effect of ablating the key components of our system: (i) in order to verify that the clicks obtained from $q(\mathbf{z}|\mathbf{x})$ provide meaningful information, rather than just adding randomness, we replace the top K clicks from Section 6.2.4 with randomly sampled clicks; (ii) we replace the image-level aggregation module from Section 6.2.5 with pixel-level averaging across the images; (iii) to investigate the broader benefit from aggregating over multiple masks, we consider the setting where $K = 1$, i.e. we just prompt the model with the most likely click.

For (i), we find performance declines substantially for CVC and ISIC. As this trend does not hold for BUSI however, we postulate that for BUSI, because the

dataset is more challenging, the model is less able to approximate the human distribution over clicks. By contrast, the clicks provided by the random approach are independent and identically distributed across all possible pixels. We find this corroborates findings in other zero-shot tasks such s active learning, which finds random selection to be an effective baseline, when prior knowledge is weak (Liu et al., 2023b). For (ii), we find performance declines across all three datasets, with DSC generally being worse than even the baseline. This supports the importance of accounting for inter-pixel dependencies in the aggregation process. For (iii), we also note a considerable decline, with worse performance across the board compared to the baseline. This shows that simply relying on a single click is more likely to mislead the model, and that it is therefore important to aggregate over multiple possible clicks.

6.3.4 Qualitative Analysis

In Figure 6.2, we present several qualitative examples to illustrate how SimSAM's annotations differ visually from SAM. In the top row, we show how SimSAM is often able to repair gaps in the initial annotation, producing an image that is overall smoother and captures more of the ground-truth annotation regions. In the middle row, we show how due to low contrast images, SAM sometimes fails to identify an object entirely, due to its pixel-level probabilities falling below the classification threshold (0.5). By contrast, SimSAM's aggregation procedure prevents this failure state. For our SimSAM approach, we also considered the union of all the masks generated before the aggregating stage, i.e. the $K = 50$ masks from Equation 6.3. Specifically, if a pixel were part of the region of interest in any of the predictions, then it would be included in the union mask. This naturally results in a mask that is oversized relative to the ground-truth, but we use it as a way to qualitatively consider what annotations it is possible for the model to make. One way to see this is that it is a way to explore the model's 'dark knowledge' (see Section 2.7), which perhaps could be brought to the forefront with the correct user simulator. For instance, in the bottom row example we see how the ground-truth annotation actually includes a larger region of the tumour beyond what the more obvious edge boundaries would indicate. Although we find both models fail to explicitly capture this, the union region in the right hand column shows that some of SimSAM's samples were able to capture this.

6.4 Conclusion

In this chapter we continued towards the objective of agnosticism over the inputs and outputs of the one-to-many model set out in Section 1.1.2. We demonstrated a user-simulation-based approach that is flexible and readily applicable interactive segmentation systems. Thus, we have mitigated the text-only limitation from the previous chapters 3, 4 and 5. Although in this case our technique only requires user clicks, the concept of simulation + aggregation of multiple

predictions presented in this paper is much more generic, and so could be applied to other modalities where user clicks is replaced with some other form of uncertainty. In the next chapter, we provide the final piece to Part I of the thesis. In particular, we address how new information can be used at runtime to reduce uncertainty in one-to-many tasks (Section 1.1.3).

Chapter 7

Dynamically One-to-Many

7.1 Introduction

In this final chapter of Part I, we address the problem of how to dynamicly reduce uncertainty in one-to-many problems, through leveraging additional information about the user (Section 1.1.3). Recall that in previous chapters our representation of the user was relatively static, being represented either as a written persona (Chapter 3), a learned part of a teacher model (Chapter 4), or a separately trained classifier (Chapter 5). However, in reality, many interactive systems involve a series of interactions over time with a user. Naturally, it would be cumbersome to be required to retrain the entire classifier, every time some new information about a particular user was obtained, or to rewrite the persona, etc.. Instead, we seek to contribute an approach that can dynamicly and flexibly leverage new information to reduce uncertainty about the one-to-many problem being solved. As a test-bed we use AI-mediated communication, which includes systems such as smart reply and autocompletion. As many of these interactive systems often operate in conjunction with one another, they make an ideal target for dealing with real-time user behaviour.

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

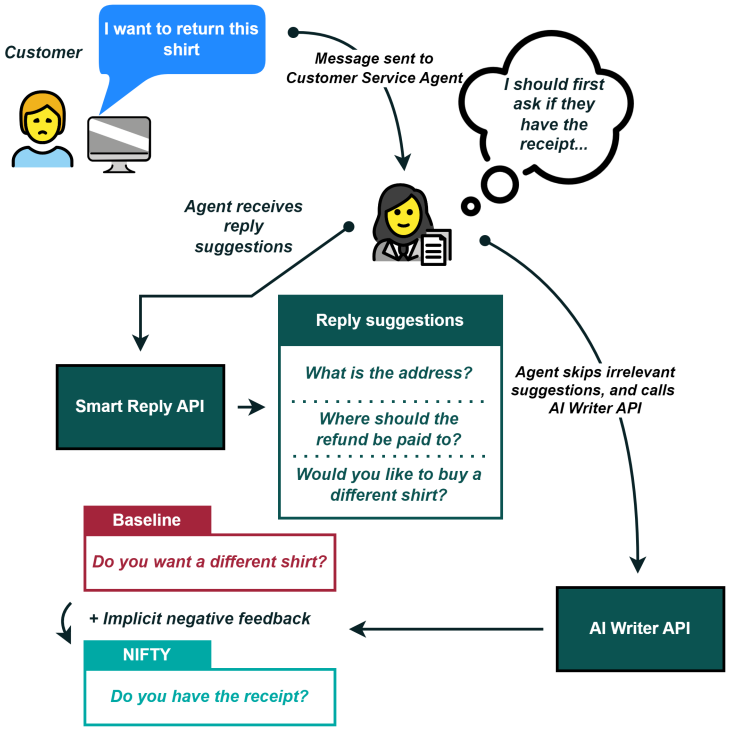


Figure 7.1: Example of how an agent may utilise either a smart reply system or an AI writing system to speed up communication with a customer. Our approach, NIFTY, uses implicit negative feedback from the rejected suggestions to improve the AI writer's prediction.

also provide an initial skeleton reply, that an AI writing model can later improve upon (Chen et al., 2019).

The interaction between these systems potentially provides information about user intentionality, allowing us to *dynamically* improve the performance by reducing the many-ness of the one-to-many problem. Yet, the heterogeneity of the forms of interaction and architectures often renders it non-obvious how the information from one system can be leveraged to improve another: e.g., a smart reply system is only useful when the user clicks one of the suggestions. In practice however, there is often too much uncertainty surrounding the user's intent for any of the suggestions to be relevant (Figure 7.1) Chakravarthi and Pasternack (2017). Due to reading the suggestions, this increases the user's cognitive load, with no additional payoff. This contrasts with positive feedback, when selecting a suggestion reduces user typing time.

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

user personalisation, although this is currently out of scope due to lack of data access.

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

Empirically, by evaluating on two publicly available datasets, we find up to 34% improvement in ROUGE-L and 89% improvement in generating the correct intent compared to a vanilla AI writing system, and an 86% win-rate according to human evaluators. In summary, our key contribution is to show how new user information can be dynamically integrated into AI-mediated communication models, enabling a reduction in the many-ness of the one-to-many problem, which was a limitation of previous chapters.

7.2 Related Work

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

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

Finally, our work builds upon the growing literature on controllable text genera-

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

7.3 Method

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

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

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

$$p_\Theta(y|x) = \prod_t^T p_\Theta(y_t|x, y_{<t}) \tag{7.1}$$

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

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

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

$$p(y_t|x, y_{<t}, c) \propto p_\Theta(y_t|x, y_{<t}) \cdot p_\Phi(c|x, y_{\leq t}) \tag{7.2}$$

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

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

$$\underset{j=1:J}{\operatorname{argmax}}\, p_\Phi(z_j|x, y_{\leq t}) \cdot \mathbb{1}_{\mathbf{z}\setminus\mathbf{z_s}}(z_j) \tag{7.3}$$

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

7.4 Experiment

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

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

Table 7.1: Results on MultiWOZ and SGD test sets using ROUGE-L and R@1 metrics. **Bold** indicates best result. * indicates result is statistically significant with p-value < 0.01 compared to best baseline.

Metrics We employ both automatic and human evaluation. ROUGE-L, which has been used previously in AI writing tasks (Ito et al., 2019), measures the longest common subsequence within the prediction, capturing surface-level overlap with the ground-truth. However, there are often many ways of expressing the same meaning that lack term-overlap. Therefore, inspired by previous evaluation work on smart reply systems (Weng et al., 2019), we complement this metric with R@1, which measures the proportion of generated responses that contain the correct intent. As the responses are generated on-the-fly, we detect intents using a DistilBERT-based classifier trained to map utterances to the corresponding intent for each dataset (a separate classifier to the one used in NIFTY), which we then compare to the ground-truth intent. We conduct human evaluation using Amazon Mechanical Turk. As the purpose of the system is to reduce uncertainty about the user's intended reply, we apply a pairwise setup, in which the evaluator is asked to *select which candidate reply is most similar to the target reply*. For each model and dataset, we randomly select 100 data points from the test set and assign 3 annotators to each data point. We define a 'win' as when a system is achieves a majority vote for a given data point. Overall, the procedure was carried out by 268 unique annotators, across 400 data points.

Baselines We compare our approach to: the standard *Baseline* AI writing model – i.e. without any classifier guidance; a *Reranker* approach which reranks the final output beams, without any token-level reranking, by selecting the beam with the highest score for the desired intent / action; an *Unlikelihood* decoding

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

Table 7.2: Results on MultiWOZ and SGD test sets under different values of α, using the T5 NIFTY Intent model.

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

Table 7.3: Human evaluation using crowdworkers. * indicates result is statistically significant with p-value < 0.01 using binomial test.

approach that downweights the probability for terms that occur in the rejected intents, encouraging the model to generate one of the non-rejected intents (Welleck et al., 2020); a *Rules-based* approach in which multiple candidate beams are generated, and are then filtered to remove beams containing rejected intents from the reply suggestions.

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

Main Results Table 7.1 presents the results for our overall system. We find our approach consistently increases the model's ability to generate the correct intent, with an improvement of up to 89% compared to the baseline. This further corresponds to an improvement in ROUGE-L of up to 34% compared to the baseline. We find that the Reranker and Rules-based approaches fail to improve much upon the Baseline approach, which is consistent with the findings of previous work (Shuster et al., 2021b), as the resulting beams typically do not represent a broad range of intents. The Unlikelihood approach also struggles due to different intents often having significant term overlap. Choice of base model proved important for some methods, although ultimately NIFTY Intent remained

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

Table 7.4: Qualitative example from the MultiWOZ test set. Text is post-processed to remove reference numbers as the model does not have access to booking API. Text in square brackets is annotated intents from dataset.

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

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

Varying Degree of Guidance In Table 7.2, we further evaluate how performance for the strongest version of our method, T5 NIFTY Intent, varies under different levels of α, which determines the weighting of the classifier on the logits, i.e., the rightmost term of Equation 7.1. We find although the approach is broadly flexible to a range of values, higher ROUGE-L scores are associated with weaker levels of guidance, while stronger levels result in higher R@1. Overall, we find the middle-ground setting of 1.0 offers the best trade-off, as well as not requiring any additional hyperparameter search.

Human Evaluation Table 7.3 shows our human evaluation results. Across both datasets and models, NIFTY Intent is statistically significantly judged better in the pairwise evaluation, with up to an 86% win rate.

Case Study Table 7.4 presents a qualitative example of model performance. By utilising the fact that the user simulator rejected the unsuccessful booking intent, NIFTY correctly surmises that a successful booking intent is instead re-

quired, in contrast to the baseline model, which refuses to make the booking. This example also supports our intuition in designing the user simulator on the assumption that the user would not select a suggestion that only partially overlapped in intent with the ground truth. In this case, although `[general-reqmore]` is present in both suggestions and target, the lack of the `[booking-book]` intent appears critical to appropriately responding to the message.

7.5 Conclusion

In this chapter, we provided the final cornerstone to Part I of the thesis, by addressing the problem from Section 1.1.3 of how to dynamicly introduce new information into the prediction process in order to reduce the uncertainty of the one-to-many problem. In contrast to the previous chapters, in which the representation of the user was primarily static, such as via a written persona (Chapter 3), a learned part of a teacher model (Chapter 4), or a separately trained classifier (Chapter 5), our contribution showed how information from one interactive system could be dynamicly introduced into another interactive system midsession in order to reduce uncertainty without requiring the model parameters to be changed.

As this concludes Part I of the thesis, we now approach Part II. Where Part I was concerned with tasks that had an implicitly one-to-many structure, but whose final output to be produced was indeed a single output, Part II is concerned with cases where the model needs to produce multiple final outputs as an end goal. This incurs with it additional challenges. Namely, (i) the problem of how to optimise for multiple outputs (Section 1.1.4), given as aforementioned in Section 2.2, machine learning by default only optimises typically for a single output, and neither datasets nor loss functions favour automatically producing multiple outputs; (ii) the problem that for some tasks the predominant architecture is deterministic (Section 1.1.5) and therefore will always produce the same output for a given input, and so we must contribute a means to transform these architecture from 'hard' one-to-one models, to one-to-many models.

Part II

Explicitly One-to-Many

Chapter 8

Optimisation for Multiple Outputs

8.1 Introduction

In this chapter, we advance to Part II of the thesis, whereupon we are now focused not only tasks that are implicitly one-to-many, but those that are explicitly one-to-many in that they require multiple outputs to be produced as the final goal. This chapter can be seen as a sister chapter to Chapter 9, as they both address the problem of how to explicitly optimise for producing multiple outputs (Section 1.1.4). To summarise the issue, machine learning by default typically revolves around predicting a single output given an input (Section 2.2). In Part I, this was not a major concern, as our goal there was often to take a highly uncertain one-to-many task, and to leverage additional information, especially about the user, such as via a written persona (Chapter 3), a learned part of a teacher model (Chapter 4), a separately trained classifier (Chapter 5), or dynamic information from another interactive system (Chapter 7), in order to **reduce** its one-to-manyness. In Part II, however, we are not focused so much on reducing its one-to-manyness, as **representing** its one-to-manyness, via producing multiple outputs that are reflective of the underlying distribution.

A natural domain to explore this is automated response suggestion, or Smart Reply (SR), which is rapidly becoming a staple feature of many email and chat systems such as Gmail, Skype, Outlook, Microsoft Teams, LinkedIn and Facebook Messenger. This is a form of interactive AI where the AI serves as an assistant to the user to help them communicate quicker. Specifically, given a message, SR systems present the user with a selection of possible responses, e.g. `How are you?` $\rightarrow$ {`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

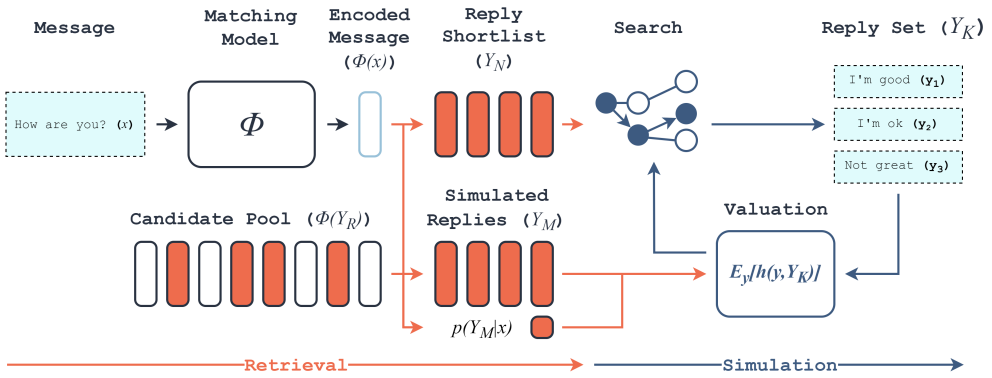


Figure 8.1: Overview of our approach. We combine a retrieval stage, which obtains the initial reply shortlist Y_N, followed by a simulation stage, which iteratively searches for reply sets Y_K from that shortlist, and evaluates their relevance against a set of simulated replies Y_M.

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., 2021b; 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 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 8.1)–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 PersonaChat–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.

8.2 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., 2021b), 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 predefined 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., 2017b; 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., 2016e) 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., 2016f). 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., 2021a). 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., 2016e)–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.

8.3 Framework

8.3.1 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., 2021b):

$$f(y) = \max_k [\{\text{sim}(y, y_k)\}_{k=1}^K] \tag{8.1}$$

8.3.2 Matching Model

Following previous approaches, we use a Matching model as the backbone of our method (Henderson et al., 2017; Zhang et al., 2021b). 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:

$$p(x_i, y_i) = \frac{\exp(g(x_i, y_i))}{\sum_{y_j} \exp(g(x_i, y_j)) + \sum_{x_j} \exp(g(x_j, y_i)) - \exp(g(x_i, y_i))} \tag{8.2}$$

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.

8.4 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., 2017b; 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:

$$\mathbb{E}_{y \sim Y}[f(Y)] \geq f(\mathbb{E}_{y \sim Y}[Y]) \tag{8.3}$$

where $f(Y)$ refers to the similarity function from Equation 8.1. The right hand side of Equation 8.3 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 2 and Figure 8.1 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 8.4.1). Then we combine a search module which selects and constructs reply tuples from this shortlist to evaluate

Algorithm 2 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

(Section 8.4.4) and a valuation module (Section 8.4.3) which computes an expected score between a given reply set and a list of simulated replies (Section 8.4.2). Note that as our method does not require learning any new parameters, it can be applied to reply sets of arbitrary sizes during inference.

8.4.1 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.

8.4.2 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.

8.4.3 Valuation

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

$$h(y_m, Y_K) = \max_k \{\text{sim}(y_m, y_k)\}_{k=1}^{K} \tag{8.4}$$

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 $y_m \in Y_M$:

$$E[h(y, Y_k)] = \sum_{m}^{M} h(y_m, Y_K) \cdot p_{model}(y_m|x) \tag{8.5}$$

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

8.4.4 Search

Given our method for estimating the value of any given K-tuple, it is necessary to employ a search 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 8.5 where the similarity function $\text{sim}(\cdot, \cdot)$ 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

Search	Reddit		PersonaChat		# 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 8.1: Results on the Reddit and PersonaChat Test sets under different search strategies for SimSR.

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 8.5.

8.5 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 8.5.5)
- **RQ2:** How does SimSR compare to existing SoTA SR methods? (Section 8.5.6, 8.5.8)

- **RQ3:** How much does SimSR benefit from accounting for interdependencies between replies when selecting a reply set? (Section 8.5.7)

8.5.1 Baselines

We identify four types of diversification strategies which serve as baselines against our model. The 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 8.3.2) (Henderson et al., 2017; Zhang et al., 2021b). 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., 2019b), whereas previous methods used a variety of transformers (Zhang et al., 2021b) 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) re-weights 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., 2017b) 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			PersonaChat		
	Train	Valid	Test	Train	Valid	Test
# Samples	50k	5k	5k	66k	8k	8k

Table 8.2: Statistics for the datasets.

8.5.2 Datasets

We evaluate our methods across two datasets, summarised in Table 8.2. 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., 2021b). We supplement this by also evaluating on PersonaChat (Zhang et al., 2018b), which similarly falls under the broader umbrella of open-domain dialogue. Below we provide further elaboration:

Reddit or MRS (Zhang et al., 2021b) 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).

PersonaChat (Zhang et al., 2018b) 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., 2020b). 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.

8.5.3 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., 2021b). This consists of a mixture of 1/2/3-grams for ROUGE-F1 (Section 2.10):

$$\frac{\text{ROUGE-1}}{6} + \frac{\text{ROUGE-2}}{3} + \frac{\text{ROUGE-3}}{2} \tag{8.6}$$

8.5.4 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., 2019b),[1] 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., 2021b). 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 PersonaChat 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.

8.5.5 Choosing a Search Strategy

Table 8.1 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

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

Section	Method	Reddit		PersonaChat	
		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 8.3: Performance of SimSR (B) compared to baseline approaches (A) and ablations (C) on the Reddit and PersonaChat Test sets. All results are statistically significant on t-test with p-value < 0.01.

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.

8.5.6 Main Results

Table 8.3A-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 PersonaChat, 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 PersonaChat is a less noisy dataset, being made by crowdworkers, and therefore both metrics are comparatively higher.

8.5.7 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 8.3C we present an ablation ('- Multi-Reply') that selects the top-K replies according to their *indi-*

PersonaChat	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...
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 8.4: Examples of model outputs on the PersonaChat (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.

vidual scores in simulation, without considering their scores at the *tuple*-level, i.e. $\mathrm{TopK}(\{\mathbb{E}[h(y, y_n)]\}_{n=1}^{N})$. We also present a version without simulation at all as a baseline comparison, which is equivalent to the Matching model in Table 8.3A.

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 PersonaChat, 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 PersonaChat 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.

8.5.8 Case Study

Table 8.4 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.

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

Table 8.5: Latency of SimSR compared to baseline approaches on the Reddit Validation set.

8.5.9 Latency

Table 8.5 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.

8.6 Conclusion

In this first chapter of Part II, we provided the first contribution towards addressing the problem of optimising for multiple outputs (Section 1.1.4). Our model-based simulation approach achieved state-of-the-art performance whilst not requiring any additional retraining. Further, given this technique only required an initial model that could produce multiple outputs and does not make any task-specific assumptions, it has the potential to be highly extensible to a range of other one-to-many tasks. In the Chapter 9 following, we continue to address the same problem, but by providing further improvements to the current approach. In particular, we address the latency of the simulation process, which is the bottleneck to the approach presented in this chapter.

Chapter 9

Efficient Optimisation for Multiple Outputs

9.1 Introduction

In this chapter, we forms the sister to Chapter 8, we continue to address the problem of how to optimise for multiple outputs (Section 1.1.4). In the previous chapter, we showed how explicitly one-to-many tasks could be optimised directly without the requirement for heuristics. However, this came at the cost of increased latency due to the requirement to perform online simulation for each prediction. Furthermore, it still required the initial subset of replies to contain a good reply set. In practice, we find Matching models (Chapter 8) often search myopically, only retrieving candidates that are very similar to one another (Figure 9.1A). Thus, the chosen reply set often fails to reflect a diverse range of user intents, while latency constraints make more sophisticated diversification techniques or larger shortlists prohibitive (Deb et al., 2019).

An intuitive, but – to the best of our knowledge – unexplored, solution to this problem is to conduct the retrieval autoregressively, with each reply conditioned on *both* the initial message *and* the previous replies in the set. Unfortunately, this approach encounters a second problem, namely, the lack of any datasets containing (message, reply set) pairs (Towle and Zhou, 2023b). In practice, SR systems are trained on individual (message, reply) pairs obtained from conversation datasets, while the task of presenting multiple diverse replies to the user is outsourced to a separate diversification module.

To meet this dual need, we present both (i) a bootstrapping method for creating a high-quality dataset of (message, reply sets) and (ii) a novel autoregressive retrieval model which predicts sequences of replies. For solving (i), we observe how model-based planning algorithms have been known to serve as a powerful policy improvement operator (Silver et al., 2017; Schrittwieser et al., 2019), including in several NLP systems (Jang et al., 2020, 2021a). Specifically, the

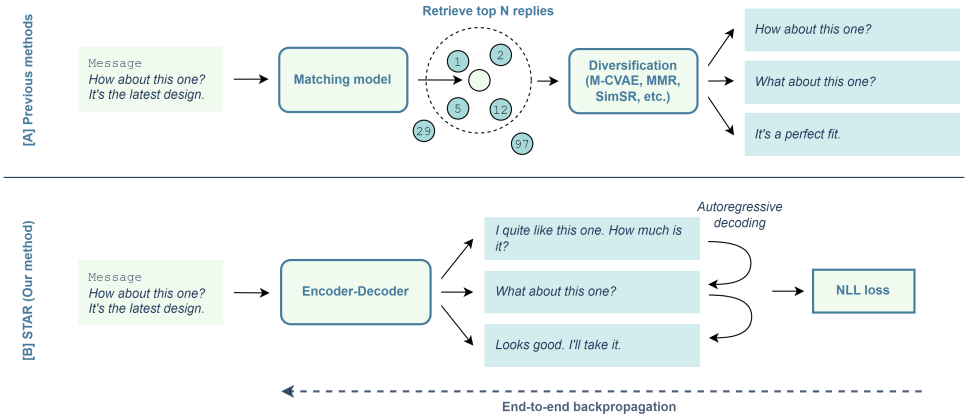


Figure 9.1: Previous methods [A] compared to our approach, STAR [B]. The example displayed is taken from the DailyDialog Test set, and compares the predictions of STAR with SimSR (Towle and Zhou, 2023b), the next best method. Our method's suggestions present a diverse range of topics/intents to drive the conversation.

outputs of a model-based planning algorithm can be used to bootstrap a SR system. Further, by conducting this planning offline we are able to leverage two key advantages: (1) the system is free of the latency constraints of online inference, and therefore can increase the search space coverage of the planning algorithm; (2) the system can leverage information that would not be available during inference, such as the ground-truth reply, to further guide the search process. For (ii) we unify both steps of the standard SR pipeline into a single end-to-end model, which mitigates the myopic search, and allows the model to learn to diversify its predictions in a principled way through gradient-based learning. To this end, we present **STAR** (**S**uggested replies with **T**5 and **A**utoregressive **R**etrieval) (Figure 9.1B). At a high level, STAR is a text-to-text model trained to output sequences of replies, where each reply is conditioned *both* on the initial message *and* the previous replies in the sequence. Concretely, we instantiate our method with the T5 pretrained model (Raffel et al., 2020). We expand T5's vocabulary by treating each reply in the candidate pool as a novel token, and demonstrate a simple-yet-effective technique for initialising the new token embeddings, which leverages the model's existing semantic priors. Notably, by treating each reply as a token, we limit the number of autoregressive decoding steps required, keeping the model's efficiency comparable to other retrieval-based methods.

Empirically, we evaluate our approach on three benchmarks: Reddit (Zhang et al., 2021b), which is the only publicly-available SR benchmark, as well as PersonaChat (Zhang et al., 2018b) and DailyDialog (Li et al., 2017b) which are both widely-used in dialogue research more broadly (Zhang et al., 2019b; Roller et al., 2020, *inter alia*), and share a similar conversational style with SR apps. We demonstrate superior performance over state-of-the-art baselines across all datasets, corresponding to a 5.1%-17.9% improvement in relevance, and a 0.5%-63.1% improvement in diversity compared to the best baseline approach. We further show comparable efficiency to previous methods, and perform a range of

ablations to motivate our design choices.

In summary, our key contribution is to remove the limitation of the previous chapter, thus having a model that both directly optimises the one-to-many problem, but also is extremely efficient.

9.2 Related Work

Smart reply The proprietary nature of data from email and chat applications has led several previous works to use publicly-available dialogue datasets (Zhang et al., 2021b; Deb et al., 2021; Towle and Zhou, 2023b) to benchmark SR methods, due to their analogous conversational nature. While early SR systems used generative models (Kannan et al., 2016), current production systems favour retrieval methods due to their greater controllability of outputs and superior latency (Deb et al., 2019). Increasing the diversity of reply suggestions is a key focus of previous work, which has been attempted by: (1) mapping replies to discrete intents / topics (Kannan et al., 2016; Chakravarthi and Pasternack, 2017; Weng et al., 2019); (2) re-weighting replies according to their similarity with other replies in the set (Carbonell and Goldstein-Stewart, 1998; Deb et al., 2019); (3) learning continuous latent variables to generate multiple queries (Zhao et al., 2017b; Deb et al., 2019); (4) using model-based simulation to iteratively search and evaluate the relevance of candidate reply sets (Towle and Zhou, 2023b). Our proposed method differs from all of these approaches in that our model learns to account for the interdependencies between replies through end-to-end back-propagation.

Autoregressive retrieval Integrating neural retrieval into the well-established paradigm of text-to-text models is of growing interest. Earlier work focuses on outputting a document ID given a query (Tay et al., 2022). Further work has extended this by considering alternate ways of representing the document IDs, such as through unique substrings (Bevilacqua et al., 2022). Another line of work has used autoregressive retrieval for the entity linking task (Cao et al., 2021a,b,c). There, the motivation is to reduce the large number of entities by relying on the text-to-text model's pre-existing vocabulary, rather than having to retrieve embeddings from a memory-intensive dense index. Our proposed method differs considerably from these previous works both in instantiation and motivation. Instantiation-wise, we generate *multiple* replies – critical to making this possible is the novel bootstrapping technique for creating the dataset of (message, reply set) pairs to train on. Motivation-wise, our goal is to be able to condition each reply on both the input message and previous replies in the set, enabling the model to learn to predict *sequences* of replies in a differentiable way.

Algorithm 3 Offline Dataset Creation. We use N=100, M=100, α=0.75 and λ=0.05 as our default setting.

Input Matching model Φ, message x, precomputed reply vectors $\{\mathbf{y_r}\}_{r=1}^{R}$, number of candidates N, number of simulations M, final reply set size K, query augmentation coefficient α, redundancy penalty λ.

Output reply set Y_K

$\mathbf{x}, \mathbf{y} \leftarrow \Phi(x), \Phi(y)$

$\tilde{\mathbf{x}} \leftarrow \alpha\mathbf{x} + (1-\alpha)\mathbf{y}$ ▷ query augmentation

$Y_N \leftarrow N\text{-}\underset{r}{\text{argmax}}(\tilde{\mathbf{x}} \cdot \mathbf{y_r})$

$Y_M \leftarrow M\text{-}\underset{r}{\text{argmax}}(\tilde{\mathbf{x}} \cdot \mathbf{y_r})$

$q(y_m|x) \propto \exp \tilde{\mathbf{x}} \cdot \mathbf{y_m}$ ▷ softmax over top-M scores

$Y_G \leftarrow \emptyset$

for $k \leftarrow 1$ to K **do**

$y_k \leftarrow \underset{n}{\text{argmax}} \sum_{m}^{M} f(Y_G^n, y_m) q(y_m|x) - \lambda f(Y_G, y_n)$

$Y_G \leftarrow Y_G \cup y_k$

end for

$Y_K \leftarrow Y_G$

return Y_K

Bootstrapping The idea of bootstrapping training data from limited resources has received significant recent interest in NLP, given the newly demonstrated few / zero-shot capabilities of many large language models (Brown et al., 2020). It has seen usage in few-shot shot text-classification (Schick and Schütze, 2021a), semantic similarity (Schick and Schütze, 2021b), tool-usage (Schick et al., 2023), retrieval (Izacard and Grave, 2021a), sequence generation (He et al., 2020), and instruction-tuning (Honovich et al., 2023; Wang et al., 2023b; Taori et al., 2023), amongst others. These techniques can also be seen as a form of knowledge distillation (Hinton et al., 2015b), except that the training typically involves predicting the exact token targets, rather than using the soft probabilities of a teacher model. Although sometimes these techniques are used as an addition to supervised learning (He et al., 2020), in our case there are no datasets containing the ideal reply sets to suggest to the user. Instead, we must bootstrap this in a more unsupervised way, by transforming a dataset of (message, reply) pairs into a dataset of (message, reply set) pairs.

9.3 Methodology

In this section, we first describe the model-based planning process used to obtain the bootstrapped dataset of (message, reply set) pairs (Section 9.3.1). Then, we show how the STAR architecture can be trained on this dataset (Section 9.3.2).

9.3.1 Offline Dataset Creation

Our goal is to transform a dialogue dataset $\mathcal{D} = \{(x, y)\}$ of (message, reply) tuples, into a dataset $\mathcal{D}* = \{(x, Y)\}$ where Y is the set of replies $\{y_k\}^K$ to be presented to the user. Algorithm 3 summarises this process. While our method is general to any arbitrary planning algorithm, we choose to instantiate our approach with a modified version of SimSR (Towle and Zhou, 2023b), from the previous chapter. As the original algorithm was designed for online inference, we make several changes to benefit the offline nature of our version, and detail the full implementation below.

The initial retrieval is conducted by a Matching model Φ that separately encodes messages and replies into a shared latent space. Given an encoded message $\mathbf{x} = \Phi(x)$, it retrieves the top N candidates from a pool of pre-computed reply vectors $\mathbf{Y_R} = \{\mathbf{y_r}\}^R$ by combining their dot product similarity with a pre-computed language-model bias – a standard component of SR systems to downweight overly specific replies (Deb et al., 2019).

$$Y_N = \textit{N}\text{-}\underset{r}{\text{argmax}}(\mathbf{x} \cdot \mathbf{y_r} + \beta \text{LM}(y_r)) \tag{9.1}$$

We then output the K-tuple $Y_i \in \binom{Y_N}{K}$ that has the highest expected similarity with the human reply, according to some similarity function $f(\cdot, \cdot)$.

$$\underset{i}{\text{argmax}}\, \mathbb{E}_{y \sim p(\cdot|x)} \Big[f(Y_i, y) \Big] \tag{9.2}$$

Given the objective in SR is for at least one of the replies to be relevant, the similarity function is defined as a maximum over the sampled reply and each of the replies in the reply set, using term-level F1-score: $\max_k \textsc{f1}(y_k, y)$.

We assume y is sampled from the ground-truth human distribution $p(\cdot|x)$. As we do not have access to the true human distribution in practice, we instead use the same Matching model q as a proxy for this, given it is trained on (message, reply) pairs. We then approximate the expectation by marginalising over the top-M most likely replies:

$$\approx \underset{i}{\text{argmax}} \sum_{m}^{M} f(Y_i, y_m) q(y_m|x) \tag{9.3}$$

In practice, it is intractable to evaluate every possible reply tuple, due to their combinatorial scaling. We therefore approximate this by greedily constructing the reply set one reply at a time. Formally, let Y_G be the set of currently selected replies, such that initially $Y_G = \emptyset$. Then, for each of $y_n \in Y_N$, we compute the expected similarity for the union of Y_G and y_n, termed $Y_G^n = Y_G \cup y_n$ for brevity:

$$\sum_{m}^{M} f(Y_G^n, y_m) q(y_m|x) \tag{9.4}$$

We repeat this process for K timesteps, each time appending the highest scoring reply to Y_G, i.e. until $|Y_G| = K$. Note that this greedy search process implicitly canonicalises the order of the replies, as selecting replies in this way causes them to be roughly ordered by individual message-reply relevance.

Adjustments

Scaling N and M The original SimSR algorithm was used only in an online setting (Towle and Zhou, 2023b). Therefore, the size of the search parameters N (number of replies in the shortlist) and M (number of simulated user replies) is kept low (15 and 25 respectively in the original paper). As we only need to run this model offline however to obtain the dataset, we find setting N and M to much larger values improves relevance (we use 100 for both), enabling both a broader search (i.e. by increasing N) and a more accurate similarity function (i.e. by increasing M).

Redundancy penalty Early testing showed that scaling the search parameters reduced diversity. We therefore introduce a redundancy penalty, which penalises the model for selecting replies that are similar to replies already in the set Y_G. This is analogous to the inter-document similarity penalty used in the maximum marginal relevance IR (information retrieval) technique (Carbonell and Goldstein-Stewart, 1998).

$$\sum_{m}^{M} f(Y_G^n, y_m) q(y_m|x) - \lambda f(Y_G, y_n) \tag{9.5}$$

Query augmentation Unlike during online inference, we also have access to the ground-truth reply y when constructing the dataset. Previous work has found that models obtain greater representational capabilities when given access to posterior information (Paranjape et al., 2022; Towle and Zhou, 2022). We therefore use an augmented query to retrieve with the Matching model. This is obtained by interpolating between the message and ground-truth reply embeddings. This biases the model's predictions towards the observed ground-truth in the dataset, while still allowing it to benefit from its own learned distribution.

$$\tilde{\mathbf{x}} = \alpha\Phi(x) + (1-\alpha)\Phi(y) \tag{9.6}$$

9.3.2 Proposed STAR Model

We initialise STAR with a T5-based text-to-text language model, which has previously been shown to be effective in autoregressive retrieval (Tay et al., 2022). While some autoregressive retrieval approaches identify their documents/replies through unique substrings (Bevilacqua et al., 2022) or constrained beam search (Cao et al., 2021b), we focus on approaches requiring only a limited number of autoregressive steps, to maintain competitive inference speeds to existing retrieval methods (Section 9.5.3). There are several alternatives for this such as treating each reply set as a unique token, or separately training on each (message, reply pair), but ultimately we opted for autoregressively treating each reply as a unique token in the vocabulary in order to exploit the compositionality of reply sets (Section 9.5.2 for performance comparison). Note that as the types of replies used in smart reply are usually quite short and concise, e.g. 'how are you', 'I'm fine thanks', 'yes, that's right' etc., systems in deployment only need to retrieve from a pool of 30k or so replies (Deb et al., 2019), in order to provide good coverage of possible user intents. As a result, we are able to keep the size of the vocabulary reasonable. Thus, our new vocabulary is defined as: $W_{tokens} \cup W_{replies}$. An obvious challenge to this approach is that by treating each reply as a previously unseen word, it removes any semantic priors the model might have about their meaning. To mitigate this, we employ a **bag-of-words** initialisation strategy. Hence, we define the embedding of the r-th reply $E(y^{(r)})$ as the average over the embeddings of the individual words within $y_t^{(r)} \in y^{(r)}$.

$$E(y^{(r)}) = \frac{1}{T} \sum_{t}^{T} E(y_t^{(r)}) \tag{9.7}$$

Intuitively, this ensures that the initial embeddings are close to the word embeddings of the original vocabulary, while also capturing some of the underlying semantics of the reply. We allow the weights to update during fine-tuning. Note that for T5 the output and input embedding layers share weights, and therefore this approach is used to initialise both layers. We train the model using cross-entropy loss to predict the next reply given the current sequence of replies and messages:

$$\mathcal{L}_{NLL} = -\sum_{k}^{K} \log p(y_k | x, y_1, ..., y_{k-1}) \tag{9.8}$$

9.4 Experimental Setup

9.4.1 Baselines

Previous work has largely been closed-source and is therefore unavailable for direct comparison (Henderson et al., 2017; Weng et al., 2019; Deb et al., 2019).

With the exception of SimSR, which has publicly available code [1], we re-implement a variety of methods that cover the broad range of previous techniques. Due to its comparable size, all baselines apart from Seq2Seq are initialised with DistilBERT as the encoder backbone. These are summarised as follows:

Seq2Seq is a generative encoder-decoder. While current production systems and the majority of related works use only retrieval models (Deb et al., 2019; Towle and Zhou, 2023b), at least one related work includes a standard generative transformer as a baseline (Zhang et al., 2021b), which we follow here. For maximum comparability with our method, we use the same `t5-small model` as a backbone. For each message, we sample K responses independently.

Matching represents the out-of-the-box encoder with no additional diversification strategy and was used as a baseline method by Zhang et al. (2021b). It simply selects the top K responses according to individual message-reply scores.

Matching-Topic uses an out-of-the-box topic classifier to ensure no two replies share the same topic, similar to previous work (Henderson et al., 2017; Weng et al., 2019). The classifier is trained on Twitter (Antypas et al., 2022), due to their comparable short-form open-domain chat conversations.

Maximum Marginal Relevance (MMR) (Carbonell and Goldstein-Stewart, 1998) is originally an IR technique, used in several previous SR works (Deb et al., 2019; Towle and Zhou, 2023b), which re-weights reply scores as a linear combination of their message-reply and inter-reply similarity.

MCVAE (Deb et al., 2019) is a conditional variational autoencoder (Zhao et al., 2017b) which learns to generate multiple query vectors from a single message embedding, representing the multiple possible reply intents. Candidates are scored via a voting process, whereby the K most-selected replies are chosen.

SimSR (Towle and Zhou, 2023b) uses an iterative search and evaluation process to select possible reply sets and score them according to their expected similarity from a learned world model, which serves as a proxy for the user. To ensure comparability of SimSR with our method and the other baselines, we include the language-model bias in the scoring process (Equation 9.1), and also deduplicate the candidate pool.[2]

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

[2] Both changes lead to consistently improved accuracy and diversity across all datasets compared to the original paper.

	Train	Valid	Test	$\lvert\mathbf{Y_R}\rvert$
Reddit	50k	5k	5k	48k
PersonaChat	66k	8k	8k	64k
DailyDialog	76k	7k	7k	62k

Table 9.1: Number of samples in the Train, Validation, Test sets and Candidate pool in the three datasets for evaluation. The Candidate pool comprises the Train set with duplicate responses removed.

9.4.2 Datasets

We evaluate our proposed method across three datasets, summarised in Table 9.1. Below, we describe the datasets in more detail and motivate their inclusion. Note, other than Reddit, there are no publicly available SR datasets, due to their commercial nature (e.g. Henderson et al. (2017); Deb et al. (2019); Weng et al. (2019)). Therefore, we adopt several dialogue datasets, which is the closest alternative to conversations on proprietary chat applications.

Reddit (Zhang et al., 2021b) was originally introduced for training multilingual SR systems, and is the only publicly available dataset specifically intended for SR purposes. As the original dataset is very large, we follow Towle and Zhou (2023b) and use the reduced version of the dataset. Note, this version only contains English, as our aim is limited to the monolingual setting. Due to the organic nature of the dataset, conversations cover a very broad range of topics.

PersonaChat (Zhang et al., 2018b) is a crowdworker-sourced dataset comprising persona-grounded conversations, in which each speaker is assigned a persona comprising a few short sentences. Following previous methods (Humeau et al., 2020b), we concatenate the persona to the beginning of the message. The participants are instructed to chat naturally and to try to get to know one another.

DailyDialog (Li et al., 2017b) is a dataset created from English language learning websites and consists of a variety of high-quality dialogues in everyday scenarios. The dataset differs from the former two in that the conversations often involve real-life scenarios, such as asking for directions, and therefore captures a different variety of conversational skills.

9.4.3 Metrics

We evaluate our method on the same weighted ROUGE ensemble as previous methods (Lin, 2004; Deb et al., 2019, 2021), which is known to correlate well

with click-through rate (Zhang et al., 2021b):

$$\frac{\text{ROUGE-1}}{6} + \frac{\text{ROUGE-2}}{3} + \frac{\text{ROUGE-3}}{2} \tag{9.9}$$

As the goal of SR systems it to ensure that at least one of the suggested replies is relevant to the user, we only record the maximum ROUGE score across each of the $K = 3$ suggested replies. We also evaluate the model on Self-ROUGE (Celikyilmaz et al., 2020): This is an unreferenced metric that measures the internal dissimilarity (i.e. diversity) within the reply set by treating one reply as the predicted reply and the other parts as the references. Note that a lower Self-ROUGE score indicates *more* diversity.

9.4.4 Inference

For inference, we use the entire training set as the candidate pool for each respective dataset, with deduplication to remove exact matches. For STAR, we greedily decode the next reply token until K tokens have been decoded. Note, we only allow the model to output replies represented in the bootstrapped dataset, and also block non-replies, i.e. words from the original vocabulary, from being predicted.

9.5 Results

We focus our efforts on answering the following Research Questions: $(\mathbf{RQ_1})$ How does STAR compare to existing state-of-the-art methods? (Section 9.5.1, 9.5.4); $(\mathbf{RQ_2})$ Which components of the data collection algorithm and fine-tuning have the largest impact on STAR's performance? (Section 9.5.2); $(\mathbf{RQ_3})$ How efficient is STAR in inference? (Section 9.5.3)

9.5.1 Main Results

Table 9.2 compares the performance of different SR systems across the Reddit, PersonaChat and DialyDialog datasets. In terms of relevance (ROUGE), STAR shows an especially large improvement in Reddit (+17.9%) and DailyDialog (+15.8%). We hypothesise the gains in PersonaChat (+5.1%) are more modest because the replies are more easily predicted due to the persona, which is concatenated to each message. This significantly reduces the noise during the initial retrieval for the baselines, as they only need to retrieve the messages relevant to that particular persona.

Method	Reddit		PersonaChat		DailyDialog	
	ROUGE ↑	Self-RGE ↓	ROUGE ↑	Self-RGE ↓	ROUGE ↑	Self-RGE ↓
Generative models						
Seq2Seq	<u>2.41</u>	3.43	6.83	**6.88***	4.01	<u>3.91</u>
Retrieval models						
Matching	1.95	9.42	7.51	21.47	6.53	16.65
M-Topic	1.81	3.94	7.16	15.43	6.14	11.11
M-MMR	2.20	4.44	7.81	14.57	6.13	8.63
M-CVAE	2.30	5.02	7.43	12.21	6.78	10.49
SimSR[3]	2.79	<u>2.18</u>	<u>9.04</u>	10.52	<u>6.82</u>	4.80
STAR	**3.29***	**2.17**	**9.50***	<u>7.74</u>	**7.90***	**1.77***

Table 9.2: Performance of STAR across Reddit, PersonaChat and DailyDialog Test sets on relevance (ROUGE) and diversity (Self-ROUGE) metrics. **Bold** indicates best result, <u>underline</u> indicates second-best. * = statistically significant versus next best result on t-test with p-value < 0.01.

Model	Reddit		PersonaChat		DailyDialog	
Configuration	ROUGE ↑	Self-RGE ↓	ROUGE ↑	Self-RGE ↓	ROUGE ↑	Self-RGE ↓
STAR	**3.35***	2.27	8.85	7.48	<u>8.39</u>	<u>1.81</u>
Data Collection Ablations						
A: No Query Augmentation	2.94	<u>2.00</u>	<u>8.99</u>	<u>6.94</u>	7.24	2.89
B: No Redundancy Penalty	<u>3.06</u>	4.29	**9.03**	17.26	**8.98***	5.90
STAR Training Variants						
C: Random embeddings	2.67	4.93	8.39	10.97	6.84	4.45
D: Reply sets as tokens	2.85	**1.59***	8.76	**6.81**	7.75	**1.57***
E: Predict replies separately	2.20	26.61	8.07	30.98	6.43	20.50

Table 9.3: Performance of STAR on the Reddit, PersonaChat and DailyDialog Validation sets under different model configurations. Ablations are applied separately. **Bold** indicates best result, <u>underline</u> indicates second-best. * = statistically significant versus next best result on t-test with p-value < 0.01.

For diversity (Self-ROUGE), the strongest gains were found in DailyDialog (+63.1%). For PersonaChat, STAR performs much better than the retrieval methods, only falling behind Seq2Seq, due to its altogether noisier outputs as evidenced by having the worst relevance score. The Reddit results were comparatively more modest (+0.5%) – we hypothesise this is because the dataset is altogether more noisy, and so there are relatively few similar replies in the dataset, as shown by the Self-ROUGE scores being lower than the other two datasets. Overall, the consistent outperformance in both relevance and diversity metrics supports the benefits of the STAR approach.

9.5.2 Ablation

In Table 9.3, we conduct ablations across two key axes: data collection and STAR training. The data collection ablations serve to investigate the benefits of the novel changes to the SimSR algorithm from Section 9.3.1. The STAR training ablations investigates the degree to which the improvements in performance are



Figure 9.2: Comparison of overall relevance and diversity scores across ablations, obtained by averaging across all three datasets with equal weighting.

caused by the bootstrapped dataset or by STAR's architecture itself; we achieve this by considering several alternative variants of STAR.

Our data collection ablations consider two features: (A) removing the query augmentation prevents the model from leveraging any ground truth information during prediction; (B) removing the redundancy penalty no longer explicitly penalises lack of diversity in predicted reply sets. For STAR training, we consider three alternative configurations: (C) we replace the bag-of-words embeddings with randomly initialised embeddings – this removes any priors about the meaning of replies and forces the model to learn them *tabula rasa*; (D) we treat each reply set as a unique token – this removes the compositional element from the task, constraining the model to only predicting previously seen reply sets, therefore testing whether the model is capable of learning to compose novel reply sets; (E) we remove the ability to account for interdependencies between replies, by restructuring each (message, reply set) data point into K data points of (message, reply$_k$), and then outputting the top-K replies during inference – this investigates whether the benefit lies simply in the bootstrapped dataset being better suited to the SR task, rather than in STAR's ability to account for interdependencies between replies.

In terms of data collection ablations, we found removing the redundancy penalty significantly reduced the diversity of predictions, although in some cases offered slightly improved relevance; removing the query augmentation generally led to a worse relevance/diversity trade-off. For the variants of STAR training, we found that random embeddings consistently reduced relevance, while also led to less diverse predictions; reply sets as tokens led to the most competitive variant of STAR compared to our default setup: diversity was overall better, due to using preconstructed reply sets from the offline planning algorithm, but this came at the trade-off of reduced flexibility from being unable to construct novel reply sets when the context required it – resultantly, we saw a corresponding reduction in relevance. Finally, predicting replies separately expectedly harmed both relevance and diversity, demonstrating the importance of accounting for reply interdependencies.

In Figure 9.2, we further validated the individual results of our ablation by aggregating the results across datasets (applying an equal weighting to each dataset). This demonstrates the overall trend that the default STAR offers the superior

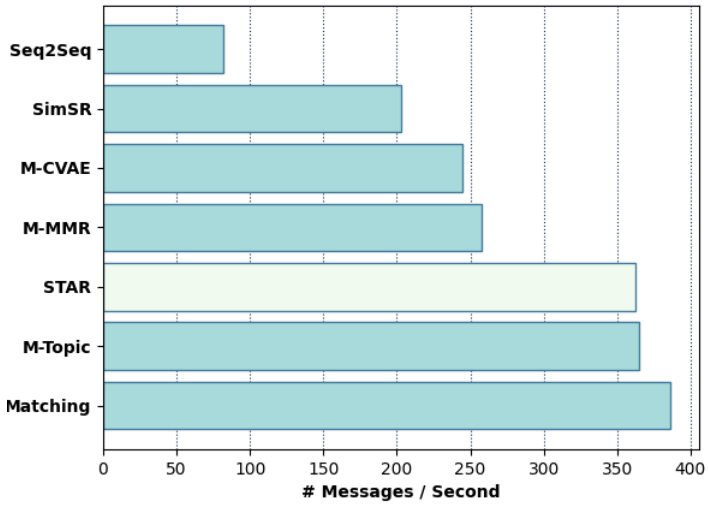


Figure 9.3: Comparison of run-time efficiency between STAR and the baseline methods. Results are calculated over the Reddit Validation set.

trade-off between relevance and diversity, while treating reply sets as tokens offered the next best alternative. Nevertheless, we believe that keeping individual replies as tokens – thus allowing the model to construct reply sets dynamically – is likely to be an attractive property for deployed systems, enabling the overall vocabulary size to remain modest.

9.5.3 Run-time Efficiency

Beyond performance gains in relevance and diversity, a major advantage of an autoregressive retrieval model is the ability to leverage the scalability of GPU-based inference. Figure 9.3 compares the efficiency of STAR with the other baseline methods. We use an NVIDIA GeForce RTX 3060 Ti GPU and AMD Ryzen 7 5700G with Radeon Graphics CPU, with a batch size of 32. The results show that the methods can be broadly clustered into three groups. The slowest group is the generative method Seq2Seq, due to needing to generate each reply word-by-word. The middle group – SimSR, M-CVAE and M-MMR – is characterised by methods that comprise a more involved diversification pipeline. The final and fastest group includes STAR, M-Topic and Matching, where no additional post-hoc diversification is required (for M-Topic the topics can be pre-computed prior to inference).

Message:	*Hi , Kenny . Let's go for a drink .*
SimSR	- let's go ! **[#9]** - ok , let's go . **[#3]** - ok . let's get something to drink . **[#1]**
STAR	- ok . let's go . **[#5]** - you want something to drink ? **[#89]** - good idea . **[#105]**
Message:	*Of course ! Let's go .*
SimSR	- let's go ! **[#1]** - ok , let's go . **[#5]** - all right . let's go . **[#12]**
STAR	- let's go ! **[#1]** - where are we ? **[#43]** - good idea ! **[#85]**

Table 9.4: Example model outputs from the DailyDialog Test set, comparing STAR (ours) with the top-performing baseline method. Numbers in **bold** indicate the ranking the reply received according to the Matching model.

9.5.4 Case Study

Table 9.4 presents a case study on the DailyDialog Test set. We compare our approach, STAR, with the top-performing baseline from Table 9.2, SimSR. In both examples we consistently find STAR is able to output a broader range of intents. Quantitatively, we consider the rank that each suggestion receives according to the initial retrieval of the Matching model that underlies SimSR. We see that STAR is able to perform a much more global search across the reply space, selecting replies from within the top 100 or so ranks. This would be difficult for the standard retrieve-and-rerank approach to emulate, given 100 is usually too large a number to efficiently rerank (Deb et al., 2019). Qualitatively, SimSR's suggestions converge around common phrases, e.g. 'let's go', which would be difficult to deduplicate with a heuristic rule given only a limited number of overlapping words between the replies. Conversely, STAR is able to represent a broader range of intents, such as replying with a question in both examples.

9.6 Conclusion

In this chapter we provided the second part to addressing the problem of optimising multiple outputs (Section 1.1.4), following Chapter 8. We introduced **STAR**, an autoregressive retrieval system for SR, which is an end-to-end text-to-

[3]Results surpass reported numbers in original paper due to inclusion of language-model bias and deduplicated candidate pool to support better comparability (Section 9.4.1).

text model that sequentially predicts replies conditioned on an initial message. The aim of this, was to alleviate the limitation pertaining to latency from the previous chapter.

To train STAR, we demonstrated an approach to bootstrap a dataset of high-quality (message, reply set) pairs, from regular dialogue datasets containing only (message, reply) pairs. Empirically, our results showed significant improvement over existing state-of-the-art SR baselines, across multiple datasets, corresponding to a 5.1%-17.9% improvement in relevance, and a 0.5%-63.1% improvement in diversity compared to the best baseline approach.

Yet, this approach, alongside the all the previous body chapters assumed some form of stochastic model that we could sample from, which enabled multiple outputs to be produced. Even in Part I, where producing multiple outputs was not the goal, the models were either large language models (Section 2.5) and so produced a probability distribution over tokens, or could produce a probability distribution over replies or actions through vector-based similarity in the case of Chapters 3 and 4. Even in Chapter 6, where we addressed the different modality of medical image segmentation, the stochasticity came via the user simulation aspect, enabling the generation of multiple annotations. In the final body chapter, we will consider the setting that is perhaps most inimical to the one-to-many problem, the case where the model itself is deterministic and yet the task requires us to produce multiple diverse and relevant outputs (Section 1.1.5). Though we will once again approach this from the perspective of medical image segmentation, this time we will not avail ourselves of the ability to inject user interaction via clicks and will consider the more challenging setting in which by default we simply have a segmentation model that produces a fixed output for a fixed input.

Chapter 10

Deterministic Models

10.1 Introduction

This chapter forms the final body chapter and also concludes Part II of the thesis. In this we address arguably the most challenging and inimical problem to one-to-many tasks, that of deterministic models (Section 1.1.5). All of the previous chapters dealt with models that were stochastic in some way, either because they were large language models (Section 2.5) and so produced a probability distribution over tokens, or could produce a probability distribution over replies or actions through vector-based similarity in the case of Chapters 3 and 4, or even in Chapter 6, where the stochasticity came via the simulation of user clicks. In some architectures however, the model is inherently deterministic and only produces a single output for a given input. A core example of this is medical imaging, where models typically produce a single segmentation per input. Although the clicks in Chapter 6 created stochasticity, there are other types of interaction that do not inherently lend themselves towards stochasticity, such as user-provided bounding boxes. It is this more challenging scenario that we consider in this chapter, where we aim to take a deterministic model and make it produce multiple outputs. As in Chapter 6, we start with the pre-trained SAM (Section 2.6) model, in order to keep the spirit of solving the problem of ensuring our one-to-many solution is relatively agnostic to underlying backbone model (Section 1.1.2).

As aforementioned in Section 2.1, the aim of this field is to annotate the boundaries of some anatomical structure such as a tumour or lesion (Ma et al., 2023). Similar to the scenarios found in Chapters 5 and 7, increasingly, pre-trained image segmentation models are being applied to the field, being a powerful and flexible tool for segmenting images, which does not rely on domain-specific bespoke models (Kirillov et al., 2023).

However, medical images contain large amounts of uncertainty. This often leads to disagreement over the 'correct' annotation for a given image, even among experts, due to confounding factors such as unclear boundaries of the structure

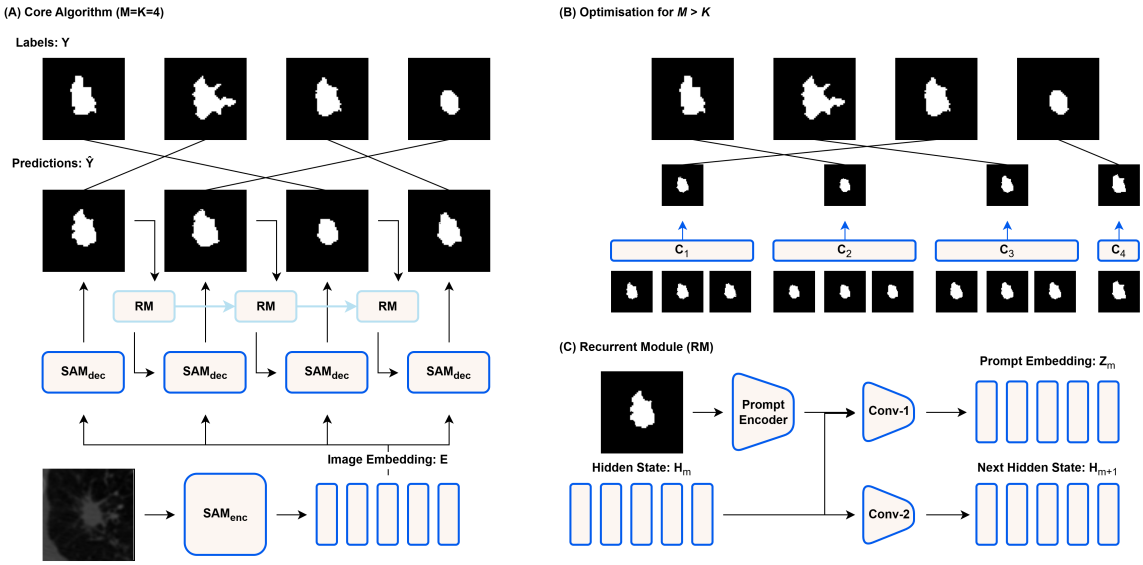


Figure 10.1: Overview of SeqSAM. A Recurrent Module sequentially generates masks. When more predictions than labels are generated, we chunk the predictions into groups and perform subsampling.

of interest or errors introduced by the annotation tool used (Baumgartner et al., 2019; Monteiro et al., 2020). This leads to an inherently interactive paradigm, as the AI can generate multiple images that can guide the diagnostic decisions of a downstream clinician.

Multiple Choice Learning (MCL) was introduced as a method to generate multiple masks for a given image (Guzmán-Rivera et al., 2012), being most recently popularised as part of the SAM architecture, a large pre-trained segmentation model (Kirillov et al., 2023). Specifically, under this approach, multiple prediction heads separately predict an image mask, and only the best image mask is backpropagated through. In medical imaging, the practical application of these techniques are numerous: MCL enables an intuitive and rapid form of semi-automatic annotation, by allowing a clinician to select from the most accurate annotation among a set of hypotheses; further, it is aligned with the collective intelligence-based approach reflected in the modern practice of case conferences and tumour boards (Rahman et al., 2023), which may alert a clinician to alternative hypotheses for a given segmentation. Indeed, the decision of where to draw the contour for a tumour can have significant medical consequences, such as in determining how invasive a surgery is (Monteiro et al., 2020). Yet, MCL has several challenges in a medical setting. In particular, MCL is limited to the fixed number of outputs it is trained on, which may fail to represent the full range of possible annotations, which often comprise subtle differences in contours. Further, the sparse winner-takes-all loss function makes it easy for one prediction head to become overly dominant, thus not guaranteeing the clinical relevancy of each mask produced.

Drawing inspiration from recurrent neural networks, we introduce SeqSAM, which sequentially generates multiple masks via a single prediction head. We

further implement a set-based optimisation procedure to enable credit assignment and ensure the clinical relevancy of each mask. Finally, we extend our approach to an arbitrary number of masks using randomised strided pooling. We evaluate on two popular multi-annotator datasets: LIDC-IDRI (Armato et al., 2011) and QUBIQ Kidney (Bran et al., 2024), and demonstrate notable improvements in quality the multiple masks produced. In summary, our main contribution is to show how even a deterministic model can be trained to produce multiple relevant and diverse outputs, thus removing the last remaining challenge for the one-to-many problem.

10.2 Methodology

10.2.1 Task Definition.

We begin with the dataset $\mathcal{D} = \{\langle \mathbf{x}_i, \mathbf{Y}_i \rangle\}_i^I$ where the i-th sample consists of an image of J pixels $x_i^{(N)} \in \mathbf{x}_i$ and a set of K plausible binary masks $\mathbf{y}_i^{(k)} \in \mathbf{Y}_i$. Our goal is to learn the function that maps the input to M outputs, where the prediction for the i-th sample is defined as the set of M hypotheses $\hat{\mathbf{y}}_i = \{\hat{\mathbf{y}}_i^{(1)}, ..., \hat{\mathbf{y}}_i^{(M)}\}$. Note that M does not have to equal K, and we demonstrate approaches for dealing with $M! = K$ in Section 10.2.5.

10.2.2 Segment Anything Model.

SAM (Kirillov et al., 2023) – the backbone of our method and baseline comparison – is a transformer-based segmentation model pre-trained on over 1 billion image masks from 11M images (see Section 2.6). The model enables various forms of user interaction such as bounding boxes, clicks and masks (sc. 'prompts') and comprises a large image encoder $\mathrm{SAM}_{\mathrm{enc}}$, with a lightweight mask decoder $\mathrm{SAM}_{\mathrm{dec}}$ and prompt encoder. Because the interaction information is only injected at a late stage in the model pipeline, multiple masks can be generated and adapted rapidly by changing the interaction inputs to the decoder, without requiring the original image to be re-encoded. At the final output layer of the model is either a single prediction head or a multi-output prediction head, depending on whether the model is being used to generate a single or multiple outputs. For the MCL baseline, we use the multi-output prediction head, while we use the single-prediction head for our method, as multiple masks can be generated through prompting the model with different input masks.

10.2.3 Recurrent Module.

By default, SAM encodes the image $\mathbf{x}$ to obtain embeddings $\mathbf{E}$, which the decoder then transforms into a single logits mask via its single-prediction head: $\mathbf{z}_1$. By leveraging SAM's native prompt encoder we can generate the next mask, simply by feeding $\mathbf{z}_1$ as an auxiliary input into the mask decoder. However, this is a stateless process, and would prevent SAM from tracking long-term information about which masks have been generated. Therefore, we connect this process via a Recurrent Module (Sutskever et al., 2014). Specifically, we define a hidden state $\mathbf{H}_m$, initialised as zeros for $m = 1$, which maintains a memory of previously seen masks. At each timestep, we update this hidden state by performing a convolution over the channel-wise concatenation of the last hidden state and current logits mask: $\mathbf{H}_{m+1} = \text{conv-2}([\mathbf{H}_m; \mathbf{Z}_m])$. In parallel, we apply a separate convolution over the same inputs to provide sequence-aware prompt embeddings for the SAM decoder: $\mathbf{Z}_{m+} = \text{conv-1}([\mathbf{H}_m; \mathbf{Z}_m])$, as shown in Figure 10.1.

10.2.4 Set-based Optimisation.

By unrolling over M timesteps, we obtain a sequence of logits masks $\{\mathbf{z}_1, ..., \mathbf{z}_M\}$ which can be mapped into probabilities through a pixel-wise sigmoid function. However, unlike in sequence prediction tasks such as language modeling (Sutskever et al., 2014), our label set $\mathbf{Y}$ lacks canonical ordering. Therefore, naively assigning the k-th label to the m-th predicted mask is unlikely to be effective for training, as the model does not know what order the labels will appear in. This is broadly referred to as the credit-assignment problem (Zhang et al., 2019a). To mitigate this, we treat the mapping between predictions and labels as a linear assignment problem. In particular, let $\mathbf{Y}_\pi$ represent the π-th permutation of labels, from a space of Π possible permutations. Then we define our loss function as:

$$\mathcal{L}(\hat{\mathbf{Y}}, \mathbf{Y}) = \min_{\pi=1:\Pi} \sum_{m}^{M} \mathcal{L}(\hat{\mathbf{Y}}_m, \mathbf{Y}_{\pi,m}) \tag{10.1}$$

where M equals the K available labels. This can be efficiently computed in $O(N^3)$ using the Hungarian algorithm (Carion et al., 2020). As each prediction is assigned to exactly one label, it has the advantage of enforcing that the model captures the range of possible modes represented in the labels. For our choice of loss function, we use DICE loss, following previous work (Ma et al., 2023) (Section 2.3). Note that these logits are backpropagated through time during training, encouraging the logits to track information about previous masks predicted in the sequence.

				LIDC-IDRI		QUBIQ Kidney	
A	Base	Method	M	$D_{avg}^{\uparrow}$	$GED^{\downarrow}$	$D_{avg}^{\uparrow}$	$GED^{\downarrow}$
	SAM Kirillov et al. (2023)	MCL	3	82.4	.284	89.2	.257
		SeqSAM	3	**83.8** †	.245 †	**90.0**	.241
		SeqSAM	10	83.5	**.234** †	89.4	**.235**
	MedSAM Ma et al. (2023)	Deterministic	1	84.0	.344	89.3	.297
		SeqSAM	3	83.9	.248 †	89.3	.265
		SeqSAM	10	**84.2**	**.213** †	**89.9**	**.210**
B		Ablations					
	SAM Kirillov et al. (2023)	Random K	10	83.0	.237	87.6	.285
		First K	10	78.7	.308	87.7	.238
		No Backprop.	3	83.2	.255	89.8	.236

Table 10.1: Results on LIDC-IDRI and QUBIC Kidney test sets. (A) Performance of SeqSAM versus SAM MCL baseline. **Bold** indicates best result. † indicates the result is significantly better than the baseline on Wilcoxon Signed Rank Test ($p < 0.01$). (B) Effect of various ablations on SeqSAM.

10.2.5 Generating Different Numbers of Masks.

The downstream inference use case may require the model to generate a different number of masks M than the number of labels (K) per sample in the dataset. When the number of masks to be generated is less than the number of labels, a straightforward approach we employ is to simply train using the available number of labels, and then to truncate the generation during inference.

However, when the number of masks to be generated is larger than the available number of labels, forcing the model to generate additional masks beyond the sequence length it was trained on may lead to out-of-distribution issues.

To mitigate this, we adopt a randomised strided pooling approach, as shown in Figure 10.1B. In particular, we first generate a sequence of logits $\{\mathbf{z}_1, ..., \mathbf{z}_M\}$ as before, equal to the desired sequence length M. We then divide the sequence into K chunks, where each chunk $\mathbf{C}_k \in \{\mathbf{C}_1, ..., \mathbf{C}_{K-1}\}$ has the size $\lceil \frac{M}{K} \rceil$, and the final chunk $\mathbf{C}_K$ has the size $M \mod K$. During training, we then randomly sample a mask from each chunk, to obtain K masks, and perform the same optimisation as before (Equation (10.1)). During inference, we omit the sampling process, and use all of generated masks. Intuitively, this encourages better credit assignment for the model, as each chunk can become specialised for a particular role.

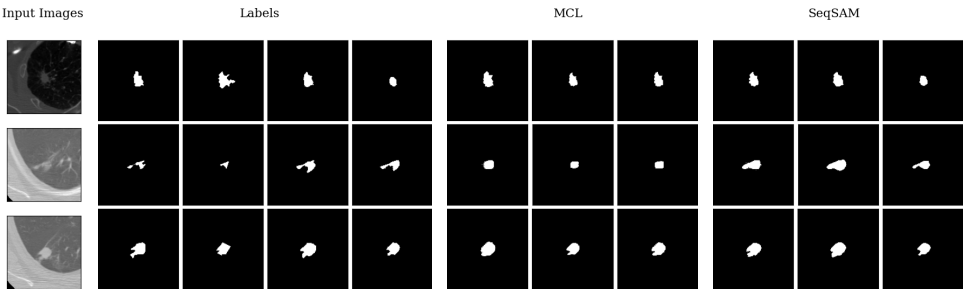


Figure 10.2: Qualitative results on LIDC-IDRI test set using SAM, showing annotated lung nodules. (Col. 1) Input images; (Col. 2-5) labels from multiple human annotators; (Col. 6-8) MCL baseline (Kirillov et al., 2023); (Col. 9-11) SeqSAM, our method, for $M = 3$.

10.3 Results

10.3.1 Experimental Design

Datasets. We evaluate on two multi-annotator datasets: (i) Lung Image Database Consortium image dataset (LIDC-IDRI) (Armato et al., 2011) ($K = 4$) and (ii) QUBIQ Kidney (Bran et al., 2024). LIDC-IDRI comprises 1018 3D thorax CT scans ($K = 3$). We use the preprocessed version provided by Kohl et al. (Kohl et al., 2018), which processes the data into 2D slices centred around each lung nodule. This results in a dataset of 15,096 2D images. We randomly split the dataset 80% / 10% / 10% for training, validation and testing respectively. QUBIQ Kidney contains 24 CT scans with 3 separate annotations; the task is to annotate the contours of the kidney and we use the default train / test split of 20 / 4 respectively.

Model Training and Parameter Setting. We train with a batch size of 2 using the AdamW optimiser (Loshchilov and Hutter, 2019), a learning rate of 1e-4 and weight decay of 0.01, until convergence. We froze the image encoder, to limit the number of parameters required to be updated. As SAM is designed for semi-automatic segmentation, for each image, we provided a bounding box, which previous studies found to be the most effective form of prompting (Ma et al., 2023). As each image contains multiple annotations, we achieve this by using the extremity points of a randomly selected non-empty mask from each label set. We baseline our method against a state-of-the-art (SoTA) MCL approach (Kirillov et al., 2023). We also compare using MedSAM, a deterministic variant of SAM specialised on medical images (Ma et al., 2023), which can also be extended to the SeqSAM method.

Metrics. We evaluate on both accuracy and distribution metrics. First, dice score – $\mathrm{D}_{\mathrm{avg}}$ – computes the average similarity across all predictions and labels, from first aggregating the predictions by pixel-wise majority voting, and then by averaging over dice scores with each label. Then, Generalised Energy Distance

(GED) is used to measure whether the generated masks capture the full *range* of possible modes for each image:

$$\mathrm{GED} = 2\mathbb{E}_{\mathbf{y}\sim\mathbf{Y},\hat{\mathbf{y}}\sim\hat{\mathbf{Y}}}[\mathrm{D}(\mathbf{y},\hat{\mathbf{y}})] - \mathbb{E}_{\mathbf{y}\sim\mathbf{Y},\mathbf{y}'\sim\mathbf{Y}'}[\mathrm{D}(\mathbf{y},\mathbf{y}')] - \mathbb{E}_{\hat{\mathbf{y}}\sim\hat{\mathbf{Y}},\hat{\mathbf{y}}'\sim\hat{\mathbf{Y}}'}[\mathrm{D}(\hat{\mathbf{y}},\hat{\mathbf{y}}')] \tag{10.2}$$

where $\mathrm{D}(\cdot,\cdot)$ is a distance function given as $1 - \mathrm{IoU}(\cdot,\cdot)$ and IoU (intersection-over-union) is given as: $\mathrm{IoU}(\hat{\mathbf{y}},\mathbf{y}) = \frac{|\hat{\mathbf{y}}\cap\mathbf{y}|}{|\hat{\mathbf{y}}\cup\mathbf{y}|}$.

10.3.2 Results

Main Results. Table 10.1A compares our method to the (Med)SAM baselines. When generating the same number of masks, our approach produces masks that are often more accurate, and also better capture the distribution of labels. We further validated the idea that our approach is able to generate an arbitrary numbers of masks. When setting the number of masks to 10, our model still produces masks that are equally as clinically relevant as when the number of masks is 3, demonstrating the flexibility of its use cases. This reveals that the additional masks are not simply copies of earlier masks but in fact capture new plausible annotations. Note statistical significance is harder to show for QUBIQ Kidney due to the smaller dataset size, although relative ranking of methods is aligned with LIDC-IDRI.

Ablation Study. Table 10.1B motivates our design choices with several ablations. In particular, we explore different ways of selecting the K masks during training, when the inference number of masks M is larger. We first use a random masks approach, whereby instead of splitting the M masks into chunks first, we sample purely from the sequence of M masks, thus making the sequence non-canonical. Then, we explore just taking the first K masks from the sequence. We find model accuracy deteriorates slightly when using random masks, suggesting the model struggles with credit assignment. When taking only the first K masks, we find very large declines in accuracy, suggesting it is important for the sequence length the model is exposed to during training to align with the number of masks produced during inference. Finally, we explored removing the backpropagation-through-time from the model's logits – in this case, backpropagation only occurs through the Recurrent Module's hidden state. We found this also reduced performance, indicating that the model's logits were also being used to effectively track information about previous masks through time.

Qualitative Analysis. Figure 10.2 provides qualitative examples of our model from the LIDC-IDRI test set. The labels reveal significant annotator disagreement regarding size and contours of the object of interest. However, the baseline MCL often misses this disagreement, also producing annotations that are more coarse-grained and simplistic in nature. By contrast, our approach, SeqSAM, is

able to capture more complexity and nuance in the shape, as well as highlighting points of disagreement also present in the human annotators.

10.4 Conclusion

In this final chapter for Part II and final body chapter of the thesis, we addressed the problem of how to endow one-to-many behaviour in an architecture that is inherently deterministic (Section 1.1.5). This was arguably the most challenging setting for one-to-many tasks, as all of the previous contributions introduced in both Parts I and II had assumed some form of stochastic model either because they were large language models (Section 2.5) and so produced a probability distribution over tokens, or could produce a probability distribution over replies or actions through vector-based similarity in the case of Chapters 3 and 4, or even in Chapter 6, where the stochasticity came via the simulation of user clicks. Ultimately however, we achieved this through a sequential approach, that demonstrably generated multiple diverse and clinically relevant masks *without* changing the model's deterministic nature.

Part III

Conclusion

Chapter 11

Conclusion and Future Work

11.1 Conclusion

In this thesis, we have discussed and presented solutions to the interconnected series of challenges that comprise the one-to-many problem. We split our solutions into two coarse-grained parts. First, we presented solutions on implicitly one-to-many tasks. These were tasks in which there was an inherent amount of uncertainty (i.e. multiple plausible outputs for a given input), but where the end task still required the production of a single output.

We first identified two interrelated problems. First was the problem of 'mode unavailability', whereby despite multiple plausible outputs existing for a given input, in some tasks the constraints on outputs were such that not all outputs would be available (i.e. retrieval tasks). In Chapter 3, we showed how a posterior-aware approach could effectively mitigate this issue. This prevented the model from attempting to predict non-existent options, allowing for greater accuracy and precision to predicting actually existent options. Related to this, the problem of mode collapse had previously hampered models in many one-to-many tasks. This occurred where the model failed to learn the full range of possible outputs for a given input and instead collapsed to predicting the most popular outputs. Thus in Chapter 4, we demonstrated a knowledge distillation-based approach that first learned to predict multiple outputs, then distilled this performance into a one-to-one model. While we demonstrated this in the context of text-based games, the implication would enable learning richer one-to-many distributions across a variety of tasks such as dialogue.

However, a key limitation of the above two chapters was the reliance on bespoke architectures. Given that modern machine learning has moved towards large language models, which are capable of performing a variety of tasks with limited specific training, the obvious next step was to develop one-to-many solutions compatible with these kinds of architectures. Thus in Chapter 5, we devised a classifier-guidance based approach to control the behaviour of an LLM depen-

dent on the particular user. The implications of this design were that it was agnostic to the choice of underlying architecture, making it highly flexible to future architectural changes which are likely to come.

Beyond this, in Chapter 6, we extended the flexibility of our approaches by branching out to other domains. Specifically, we considered the case of medical image segmentation, which is widely considered a one-to-many problem due to the vast amounts of uncertainty on any particular segmentation. In this case, we adopted a simulation-based approach that used the model's uncertainty values to imitate the behaviour of a human annotator and found superior performance compared to previous methods. This further showed that many of our conclusions and techniques were generalisable to other domains.

Finally, in Chapter 7, we removed a final limitation from the previous chapters. Namely, the assumption that the source of uncertainty in the interactive AI system was static. However, of course as a human interacts with a system, information about that user will build up, enabling us to reduce uncertainty. In this chapter therefore we sought to design an architecture to achieve that. Specifically, we demonstrated a classifier-guidance based approach again, which would enable any two interactive AI systems to share information about a user to improve their performance, without any architectural changes being required by either system. This is useful because often in industry separate teams will be working on each system, so an architecture that requires as little coupling as possible between the systems would be much easier to implement in practice.

In Part II of the thesis, we considered solutions for the explicit variant of the one-to-many problem, that requires actually predicting multiple outputs for a given input. A common example of this used throughout the thesis is smart reply, whereby a user receives a message and the AI provides multiple possible replies to said message, as seen in Microsoft Outlook or Google Gmail. One of the more unique challenges to this problem was that the goal is that at least one of the suggestions should be relevant to the user, but we do not care about the relevance of the 2nd or 3rd suggestion. Previous approaches had approached this through a trade-off between relevance and diversity in order to encourage the suggestions to cover a broad range of possible user replies. In Chapter 8 however, we instead developed a simulation-based approach that directly optimised the aforementioned end goal by searching across candidate reply sets and evaluating them using a user simulator. The benefit of this was that it did not require careful parameter-tuning between relevance and diversity and instead directly achieved the underlying intent of the system.

However, a major challenge of a simulation-based approach was that it took longer than regular methods to run, potentially creating challenges around latency. We sought to remedy this in Chapter 9, where we instead performed the simulation offline, and then used knowledge distillation to train a faster model to emulate the behaviour of the simulator, using a novel autoregressive retrieval model that enabled predicting replies in a sequence. In practice, we even showed greater performance than before, because running it offline meant we could gen-

erate even higher quality data to train on, compared to what an online-only latency would allow.

Finally, we solved the most challenging, lingering problem of the thesis. Namely, the problem of how to deal with deterministic architectures. This was a challenge as all of the above techniques assumed we had some probabilistic model we could sample from to generate multiple outputs. However, in some cases such as medical image segmentation, this is not the case. Instead, in Chapter 10, we transformed the model into a recurrent neural network, which enabled us to generate multiple outputs in a row, while still maintaining the model's deterministic nature. The implication of this was that we showed that even if a model does not appear to be able to generate multiple outputs, there are various tricks and techniques that can be employed to achieve this, without having to redesign the underlying model.

Overall, as we have developed solutions towards the one-to-many problem, a key theme has been flexibility. This is because while the one-to-many problem is structurally unusual as a machine learning problem, given the field is primarily geared towards one-to-one problems, this same field has also afforded a large number of powerful off-the-shelf models and architectures. As such, while our earlier work focused on bespoke solutions, we have trended towards looking at how these existing solutions can be adapted towards the one-to-many problem in order to leverage their greater expressive power. Finally, this can be seen in how our work spans both text and image modalities, to show that many of the techniques shown, while requiring some degree of adaptation, are fundamentally applicable across multiple modalities and tasks.

11.2 Future Work

There are several lines for future work to explore. First, work could build horizontally on the research here by expanding to a broader range of interactive systems, including recommender systems, internet search, video games, adaptive user interfaces, amongst others. When transitioning our work between smart reply and image segmentation tasks, we found some of our techniques, such as simulation-based approaches transferred well into both domains. However, we also found that each task presented certain distinct challenges. For instance, the medical imaging lacked the inherent stochasticity present in generative and retrieval NLP models, thus requiring additional measures in the form of click simulation, which can be parameterised in a stochastic way. We therefore expect that unique challenges may present themselves in adapting to these systems, although we expect our techniques to be broadly applicable.

The second major direction is to build vertically on the systems presented in this thesis, by improving the respective architectures, and we provide several suggestions here. For the STAR architecture (Chapter 9), solving the cold-start

problem from the new expanded vocabulary would be useful. In our version, we used bag-of-words average embeddings, which outperformed random initialisation, however, this is naturally limited by the fact that words are related to each other non-linearly, and some words in a sentence are more important than others. A second limitation is the reliance on the F1 score to simulate the relevance of a particular response. In practice, automated metrics are known to be only loosely correlated with human preferences, and it could be argued that methods such as SimSR and STAR heavily overfit the replies to trivially avoid term-level overlap. Future work may look at how to directly learn a value function from user behavior. This is not straightforward, as the desired response of the user is not directly observed when the user chooses a suggestion. Indeed, the only information we have in that case is the final message sent by the user, which is an adapted form of the suggested reply. One solution in this case may be to treat the user's intended reply as a latent variable, and to build a click prediction model that marginalises over that variable. In the case where the user rejects all suggestions, we see directly what the user wanted to say, so the similarity function could be directly optimised in that case.

Another line of work in the click simulation approach for SimSAM (Chapter 6), would be to use more sophisticated click selection techniques. While we sampled clicks from the places of highest model entropy, this is not necessarily the best place to click, as the highest points of entropy are also usually the least informative points. While this may appear counterintuitive given information theory, it is because the highest entropy points are usually directly on or around the edge of the region of interest, however these pixels have limited correlation to the pixels around them. A more sophisticated technique could consider the information gain of the click as well therefore.

Due to the rapidly advancing state-of-the-art of deep learning, there are still emergent challenges in the one-to-many problem that are beyond the scope of the current thesis. For instance, the size of many LLMs puts them out of reach of most home PCs, requiring them to be only interacted with via an API. This creates a challenge for many of the techniques in this thesis, as an API only allows for interaction via a textual prompt, preventing us from even applying classifier-guidance to the model's output logits.

A further challenge that we only touched on in Chapter 5 is how can LLMs be meaningfully personalised in a way that is scalable. Currently, LLMs only allow for manipulating the prompt, such as putting previous examples of user preferences in the prompt, but this appears to be a somewhat clunky and inefficient approach to controlling the LLM, and deviates significantly from how recommender systems are operating, which have historically been hugely successful in personalisation.

Finally, in this thesis, we have focused on the one-to-many problem as it pertains to *interactive* systems. However, that is by no means to say that it is not a problem *outside of* interactive systems. For instance, in object detection, an object may be partially occluded behind a wall. In this case, there is uncertainty about the

exact shape and nature of the object obscured by the wall. Another popular example given by machine learning researcher Yann LeCun is that of future frame prediction in video. Given a pen held perpendicular to a surface, upon release it may fall in any direction. A video generation model will typically struggle with this and predict a very faint and blurry pen falling in every possible direction, rather than committing to an outcome. Further work may thus be interested in seeing how the ideas and techniques from this thesis can be applied to such other domains.

Bibliography

Adhikary, J., Berger, J., and Vertanen, K. (2021). Accelerating text communication via abbreviated sentence input. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6574–6588, Online. Association for Computational Linguistics.

Adnan, M., Kalra, S., Cresswell, J. C., Taylor, G. W., and Tizhoosh, H. R. (2021). Federated learning and differential privacy for medical image analysis. *Scientific Reports*, 12.

Adolphs, L., Gao, T., Xu, J., Shuster, K., Sukhbaatar, S., and Weston, J. (2022). The cringe loss: Learning what language not to model. In *Annual Meeting of the Association for Computational Linguistics*.

Agarwal, R., Singh, A., Zhang, L. M., Bohnet, B., Chan, S., Anand, A., Abbas, Z., Nova, A., Co-Reyes, J. D., Chu, E., Behbahani, F. M. P., Faust, A., and Larochelle, H. (2024). Many-shot in-context learning. *ArXiv*, abs/2404.11018.

Al-Dhabyani, W. S., Gomaa, M. M. M., Khaled, H., and Fahmy, A. A. (2019). Dataset of breast ultrasound images. *Data in Brief*, 28.

Ali, S., Ghatwary, N. M., Jha, D., Isik-Polat, E., Polat, G., Yang, C., Li, W., Galdran, A., Ballester, M. Á. G., Thambawita, V. L., Hicks, S., Poudel, S., Lee, S.-W., Jin, Z., Gan, T., Yu, C.-P., Yan, J., Yeo, D., Lee, H., Tomar, N. K., Haithmi, M., Ahmed, A., Riegler, M., Daul, C., Halvorsen, P., Rittscher, J., Salem, O. E., Lamarque, D., Cannizzaro, R., Realdon, S., de Lange, T., and East, J. E. (2022). Assessing generalisability of deep learning-based polyp detection and segmentation methods through a computer vision challenge. *Scientific Reports*, 14.

Ammanabrolu, P. and Hausknecht, M. J. (2020). Graph constrained reinforcement learning for natural language action spaces. *ArXiv*, abs/2001.08837.

Ammanabrolu, P. and Riedl, M. O. (2019). Playing text-adventure games with graph-based deep reinforcement learning. *ArXiv*, abs/1812.01628.

Ammanabrolu, P. and Riedl, M. O. (2021a). Learning knowledge graph-based world models of textual environments. In *NeurIPS*.

Ammanabrolu, P. and Riedl, M. O. (2021b). Modeling worlds in text. *ArXiv*, abs/2106.09578.

Ammanabrolu, P., Tien, E., Hausknecht, M., and Riedl, M. O. (2020). How to avoid being eaten by a grue: Structured exploration strategies for textual worlds. *CoRR*, abs/2006.07409.

Ammanabrolu, P., Urbanek, J., Li, M., Szlam, A. D., Rocktaschel, T., and Weston, J. (2021). How to motivate your dragon: Teaching goal-driven agents to speak and act in fantasy worlds. In *NAACL*.

Antypas, D., Ushio, A., Camacho-Collados, J., Silva, V., Neves, L., and Barbieri, F. (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.

Armato, S. G., Mclennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., Zhao, B., Aberle, D., Henschke, C. I., Hoffman, E. A., Kazerooni, E. A., MacMahon, H., van Beek, E. J. R., Yankelevitz, D., Biancardi, A. M., Bland, P. H., Brown, M. S., Engelmann, R. M., Laderach, G. E., Max, D., Pais, R. C., Qing, D. P.-Y., Roberts, R. Y., Smith, A. R., Starkey, A., Batra, P., Caligiuri, P., Farooqi, A. O., Gladish, G. W., Jude, C. M., Munden, R., Petkovska, I., Quint, L. E., Schwartz, L. H., Sundaram, B., Dodd, L. E., Fenimore, C., Gur, D., Petrick, N., Freymann, J. B., Kirby, J., Hughes, B., Casteele, A. V., Gupte, S., Sallam, M., Heath, M. D., Kuhn, M. H., Dharaiya, E., Burns, R., Fryd, D., Salganicoff, M., Anand, V., Shreter, U., Vastagh, S., Croft, B. Y., and Clarke, L. P. (2011). The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38 2:915–31.

Arnold, M. (1861). *On Translating Homer: Three Lectures Given at Oxford*. Longman, Green, Longman, and Roberts, London.

Arora, K., Shuster, K., Sukhbaatar, S., and Weston, J. (2022). Director: Generator-classifiers for supervised language modeling. In *AACL*.

Bao, S., He, H., Wang, F., and Wu, H. (2020). Plato: Pre-trained dialogue generation model with discrete latent variable. In *ACL*.

Baumgartner, C. F., Tezcan, K. C., Chaitanya, K., Hötker, A. M., Muehlematter, U. J., Schawkat, K., Becker, A. S., Donati, O. F., and Konukoglu, E. (2019). Phiseg: Capturing uncertainty in medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*.

Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., and Blackburn, J. (2020). The pushshift reddit dataset. In *ICWSM*.

Bernal, J., Sánchez, F. J., Fernández-Esparrach, G., Gil, D., de Miguel, C. R., and Vilariño, F. (2015). Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society*, 43:99–111.

Bevilacqua, M., Ottaviano, G., Lewis, P., tau Yih, W., Riedel, S., and Petroni, F. (2022). Autoregressive search engines: Generating substrings as document identifiers. In *NeurIPS*.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K. A., Davis, J., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N. D., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T. F., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M. S., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J. F., Ogut, G., Orr, L. J., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y. H., Ruiz, C., Ryan, J., R'e, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K. P., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M. A., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. (2021). On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *CoNLL*.

Bran, H., Navarro, F., Ezhov, I., Bayat, A., Das, D., Kofler, F., Shit, S., Waldmannstetter, D., Paetzold, J. C., Hu, X., Wiestler, B., Zimmer, L., Amiranashvili, T., Prabhakar, C., Berger, C., Weidner, J., Alonso-Basant, M., Rashid, A., Baid, U., Adel, W., Ali, D., Baheti, B., Bai, Y.-L., Bhatt, I., Cetindag, S. C., Chen, W., Cheng, L., Dutand, P., Dular, L., Elattar, M. A., Feng, M., Gao, S., Huisman, H., Hu, W., Innani, S., Jiat, W., Karimi, D., Kuijf, H., Kwak, J. T., Le, H. L., Lia, X., Lin, H., Liu, T., Ma, J., Ma, K., Ma, T., Oksuz, I., Holland, R., Oliveira, A. L., Pal, J. B., Pei, X., Qiao, M., Saha, A., Selvan, R., Shen, L., Silva, J. L., Žiga piclin, Talbar, S. N., Wang, D., Wang, W., Wang, X., Wang, Y., Xia, R., Xu, K., Yan, Y., Yergin, M., Yu, S., Zeng, L., Zhang, Y., Zhao, J., Zheng, Y., Zukovec, M., Do, R., Becker, A. S., Simpson, A. L., Konukoglu, E., Jakab, A., Bakas, S., Joskowicz, L., and Menze, B. H. (2024). Qubiq: Uncertainty quantification for biomedical image segmentation challenge. *ArXiv*, abs/2405.18435.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gasic, M. (2018). Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *ArXiv*, abs/1810.00278.

Cao, N. D., Aziz, W., and Titov, I. (2021a). Highly parallel autoregressive entity linking with discriminative correction. In *Conference on Empirical Methods in Natural Language Processing*.

Cao, N. D., Izacard, G., Riedel, S., and Petroni, F. (2021b). Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Cao, N. D., Wu, L. Y., Popat, K., Artetxe, M., Goyal, N., Plekhanov, M., Zettlemoyer, L., Cancedda, N., Riedel, S., and Petroni, F. (2021c). Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290.

Carbonell, J. G. and Goldstein-Stewart, J. (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*.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. *ArXiv*, abs/2005.12872.

Celikyilmaz, A., Clark, E., and Gao, J. (2020). Evaluation of text generation: A survey. *ArXiv*, abs/2006.14799.

Chakravarthi, N. and Pasternack, J. (2017). Building smart replies for member messages. press release. https://engineering.linkedin.com/blog/2017/10/building-smart-replies-for-member-messages.

Chen, M. X., Lee, B., Bansal, G., Cao, Y., Zhang, S., Lu, J., Tsay, J., Wang, Y., Dai, A. M., Chen, Z., Sohn, T., and Wu, Y. (2019). Gmail smart compose: Real-time assisted writing. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Chen, Y.-H. and Choi, J. D. (2016). Character identification on multiparty conversation: Identifying mentions of characters in TV shows. In Fernandez,

R., Minker, W., Carenini, G., Higashinaka, R., Artstein, R., and Gainer, A., editors, *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100, Los Angeles. Association for Computational Linguistics.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Valter, D., Narang, S., Mishra, G., Yu, A. W., Zhao, V., Huang, Y., Dai, A. M., Yu, H., Petrov, S., hsin Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Dai, S., Shao, N., Zhao, H., Yu, W., Si, Z., Xu, C., Sun, Z., Zhang, X., and Xu, J. (2023). Uncovering chatgpt's capabilities in recommender systems. *Proceedings of the 17th ACM Conference on Recommender Systems*.

de Oliveira, C. M., de Moura, L. V., Ravazio, R. C., Kupssinskü, L. S., Parraga, O., Delucis, M. M., and Barros, R. C. (2023). Zero-shot performance of the segment anything model (sam) in 2d medical imaging: A comprehensive evaluation and practical guidelines. *2023 IEEE 23rd International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 108–112.

Deb, B., Bailey, P., and Shokouhi, M. (2019). Diversifying reply suggestions using a matching-conditional variational autoencoder. In *North American Chapter of the Association for Computational Linguistics*.

Deb, B., Zheng, G., Shokouhi, M., and Awadallah, A. H. (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.

Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A. S., Nemade, G., and Ravi, S. (2020). Goemotions: A dataset of fine-grained emotions. In *Annual Meeting of the Association for Computational Linguistics*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019a). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019b). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., and Weston, J. (2019). Wizard of wikipedia: Knowledge-powered conversational agents. *ArXiv*, abs/1811.01241.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., Li, L., and Sui, Z. (2023). A survey for in-context learning. *ArXiv*, abs/2301.00234.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929.

Faltings, F., Galley, M., Brantley, K., Peng, B., Cai, W., Zhang, Y., Gao, J., and Dolan, B. (2023). Interactive text generation. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4450–4468, Singapore. Association for Computational Linguistics.

Fedorenko, D. G., Smetanin, N., and Rodichev, A. (2017). Avoiding echo-responses in a retrieval-based conversation system. *ArXiv*, abs/1712.05626.

Firman, M., Campbell, N. D. F., de Agapito, L., and Brostow, G. J. (2018). Diversenet: When one right answer is not enough. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5598–5607.

Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*.

Freitas, D. D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., and Le, Q. V. (2020). Towards a human-like open-domain chatbot. *ArXiv*, abs/2001.09977.

Gao, J., Galley, M., and Li, L. (2018). Neural approaches to conversational AI. In Artzi, Y. and Eisenstein, J., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 2–7, Melbourne, Australia. Association for Computational Linguistics.

Gao, X., Lee, S., Zhang, Y., Brockett, C., Galley, M., Gao, J., and Dolan, W. B. (2019). Jointly optimizing diversity and relevance in neural response generation. In *NAACL*.

Geng, S., Liu, S., Fu, Z., Ge, Y., and Zhang, Y. (2022). Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). *Proceedings of the 16th ACM Conference on Recommender Systems*.

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

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm networks. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 4:2047–2052 vol. 4.

Gu, J.-C., Li, T., Liu, Q., Zhu, X., Ling, Z., Su, Z., and Wei, S. (2020a). Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

Gu, J.-C., Ling, Z., and LIU, Q. (2019a). Interactive matching network for multi-turn response selection in retrieval-based chatbots. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Gu, J.-C., Ling, Z., LIU, Q., Wei, S., and Zhu, X.-D. (2020b). Filtering before iteratively referring for knowledge-grounded response selection in retrieval-based chatbots. In *FINDINGS*.

Gu, J.-C., Ling, Z., Zhu, X., and Liu, Q. (2019b). Dually interactive matching network for personalized response selection in retrieval-based chatbots. In *EMNLP*.

Gu, J.-C., Liu, H., Ling, Z., LIU, Q., Chen, Z., and Zhu, X. (2021). Partner matters! an empirical study on fusing personas for personalized response selection in retrieval-based chatbots. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Guo, X., Yu, M., Gao, Y., Gan, C., Campbell, M., and Chang, S. (2020). Interactive fiction game playing as multi-paragraph reading comprehension with reinforcement learning. In *EMNLP*.

Gutman, D., Codella, N. C. F., Celebi, E. M., Helba, B., Marchetti, M., Mishra, N. K., and Halpern, A. (2016). Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic). *ArXiv*, abs/1605.01397.

Guzmán-Rivera, A., Batra, D., and Kohli, P. (2012). Multiple choice learning: Learning to produce multiple structured outputs. In *NIPS*.

Hausknecht, M. J., Loynd, R., Yang, G., Swaminathan, A., and Williams, J. (2019). Nail: A general interactive fiction agent. *ArXiv*, abs/1902.04259.

He, H., Chen, D., Balakrishnan, A., and Liang, P. (2018). Decoupling strategy and generation in negotiation dialogues. In *EMNLP*.

He, J., Chen, J., He, X., Gao, J., Li, L., Deng, L., and Ostendorf, M. (2016). Deep reinforcement learning with a natural language action space. *arXiv: Artificial Intelligence*.

He, J., Gu, J., Shen, J., and Ranzato, M. (2020). Revisiting self-training for neural sequence generation. In *Proceedings of ICLR*.

Henderson, M., Al-Rfou, R., Strope, B., Sung, Y.-H., Lukács, L., Guo, R., Kumar, S., Miklos, B., and Kurzweil, R. (2017). Efficient natural language response suggestion for smart reply. *ArXiv*, abs/1705.00652.

Hendrycks, D., Mazeika, M., Zou, A., Patel, S., Zhu, C., Navarro, J., Song, D. X., Li, B., and Steinhardt, J. (2021). What would jiminy cricket do? towards agents that behave morally. *ArXiv*, abs/2110.13136.

Hinton, G. E., Vinyals, O., and Dean, J. (2015a). Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Hinton, G. E., Vinyals, O., and Dean, J. (2015b). Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.

Honovich, O., Scialom, T., Levy, O., and Schick, T. (2023). Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hua, K., Feng, Z., Tao, C., Yan, R., and Zhang, L. (2020). Learning to detect relevant contexts and knowledge for response selection in retrieval-based dialogue systems. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

Humeau, S., Shuster, K., Lachaux, M.-A., and Weston, J. (2020a). Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *ICLR*.

Humeau, S., Shuster, K., Lachaux, M.-A., and Weston, J. (2020b). Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Irvine, R. P., Boubert, D., Raina, V., Liusie, A., Zhu, Z., Mudupalli, V., Korshuk, A., Liu, Z. J., Cremer, F., Assassi, V., Beauchamp, C.-C., Lu, X., Rialan, T., and Beauchamp, W. (2023). Rewarding chatbots for real-world engagement with millions of users. *ArXiv*, abs/2303.06135.

Ito, T., Kuribayashi, T., Kobayashi, H., Brassard, A., Hagiwara, M., Suzuki, J., and Inui, K. (2019). Diamonds in the rough: Generating fluent sentences from early-stage drafts for academic writing assistance. In van Deemter, K., Lin, C., and Takamura, H., editors, *Proceedings of the 12th International Conference on Natural Language Generation*, pages 40–53, Tokyo, Japan. Association for Computational Linguistics.

Izacard, G. and Grave, E. (2021a). Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Izacard, G. and Grave, E. (2021b). Distilling knowledge from reader to retriever for question answering. *ArXiv*, abs/2012.04584.

Jang, E., Gu, S. S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. *ArXiv*, abs/1611.01144.

Jang, Y., Lee, J., and Kim, K.-E. (2020). Bayes-adaptive monte-carlo planning and learning for goal-oriented dialogues. In *AAAI Conference on Artificial Intelligence*.

Jang, Y., Seo, S., Lee, J., and Kim, K.-E. (2021a). Monte-carlo planning and learning with language action value estimates. In *International Conference on Learning Representations*.

Jang, Y., Seo, S., Lee, J., and Kim, K.-E. (2021b). Monte-carlo planning and learning with language action value estimates. In *International Conference on Learning Representations*.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. (2020). Tinybert: Distilling bert for natural language understanding. *ArXiv*, abs/1909.10351.

Johnson, J., Douze, M., and Jégou, H. (2017). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7:535–547.

Kadlec, R., Schmid, M., and Kleindienst, J. (2015). Improved deep learning baselines for ubuntu corpus dialogs. *ArXiv*, abs/1510.03753.

Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., Corrado, G. S., Lukács, L., Ganea, M., Young, P., and Ramavajjala, V. (2016). Smart reply: Automated response suggestion for email. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *ArXiv*, abs/2001.08361.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858.

Khattab, O., Potts, C., and Zaharia, M. A. (2021). Relevance-guided supervision for openqa with colbert. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Khattab, O. and Zaharia, M. A. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Kim, B., Ahn, J., and Kim, G. (2020). Sequential latent knowledge selection for knowledge-grounded dialogue. *ArXiv*, abs/2002.07510.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. B. (2023). Segment anything. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003.

Kirillov, A., Savchynskyy, B., Schlesinger, D., Vetrov, D. P., and Rother, C. (2015). Inferring m-best diverse labelings in a single one. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1814–1822.

Kohl, S. A. A., Romera-Paredes, B., Meyer, C., Fauw, J. D., Ledsam, J. R., Maier-Hein, K., Eslami, S. M. A., Rezende, D. J., and Ronneberger, O. (2018). A probabilistic u-net for segmentation of ambiguous images. *ArXiv*, abs/1806.05034.

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S. R., Socher, R., and Rajani, N. (2020). Gedi: Generative discriminator guided sequence generation. In *Conference on Empirical Methods in Natural Language Processing*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Le, H. and Hoi, S. C. H. (2020). Video-grounded dialogues with pretrained generation language models. *ArXiv*, abs/2006.15319.

Lee, A., Auli, M., and Ranzato, M. (2021). Discriminative reranking for neural machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Lee, S., Purushwalkam, S., Cogswell, M., Ranjan, V., Crandall, D. J., and Batra, D. (2016). Stochastic multiple choice learning for training diverse deep ensembles. *ArXiv*, abs/1606.07839.

Lewis, M., Yarats, D., Dauphin, Y., Parikh, D., and Batra, D. (2017). Deal or no deal? end-to-end learning of negotiation dialogues. *ArXiv*, abs/1706.05125.

Li, C., Zhang, M., Mei, Q., Wang, Y., Hombaiah, S. A., Liang, Y., and Bendersky, M. (2023). Teach llms to personalize - an approach inspired by writing education. *ArXiv*, abs/2308.07968.

Li, F.-L., Qiu, M., Chen, H., Wang, X., Gao, X., Huang, J., Ren, J., Zhao, Z., Zhao, W., Wang, L., Jin, G., and Chu, W. (2017a). Alime assist: An intelligent assistant for creating an innovative e-commerce experience. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2016a). A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, W. (2016b). A diversity-promoting objective function for neural conversation models. *ArXiv*, abs/1510.03055.

Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, W. (2016c). A persona-based neural conversation model. *ArXiv*, abs/1603.06155.

Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, W. B. (2016d). A persona-based neural conversation model. *ArXiv*, abs/1603.06155.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. (2016e). Deep reinforcement learning for dialogue generation. In *Conference on Empirical Methods in Natural Language Processing*.

Li, T., Mesbahi, Y. E., Kobyzev, I., Rashid, A., Mahmud, A., Anchuri, N., Hajimolahoseini, H., Liu, Y., and Rezagholizadeh, M. (2021). A short study on compressing decoder-based language models. *ArXiv*, abs/2110.08460.

Li, X., Lipton, Z. C., Dhingra, B., Li, L., Gao, J., and Chen, Y.-N. V. (2016f). A user simulator for task-completion dialogues. *ArXiv*, abs/1612.05688.

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., and Niu, S. (2017b). Dailydialog: A manually labelled multi-turn dialogue dataset. In *International Joint Conference on Natural Language Processing*.

Lian, R., Xie, M., Wang, F., Peng, J., and Wu, H. (2019). Learning to select knowledge for response generation in dialog systems. *ArXiv*, abs/1902.04911.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Liu, H., Li, H., Yao, X., Fan, Y., Hu, D., Dawant, B. M., Nath, V., Xu, Z., and Oguz, I. (2023a). Colossal: A benchmark for cold-start active learning for 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*.

Liu, H., Li, H., Yao, X., Fan, Y., Hu, D., Dawant, B. M., Nath, V., Xu, Z., and Oguz, I. (2023b). Colossal: A benchmark for cold-start active learning for 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*.

Liu, Q., Chen, Y., Chen, B., Lou, J.-G., Chen, Z., Zhou, B., and Zhang, D. (2020). You impress me: Dialogue generation via mutual persona perception. *ArXiv*, abs/2004.05388.

Liu, Q., Zheng, M., Planche, B., Karanam, S., Chen, T., Niethammer, M., and Wu, Z. (2022). Pseudoclick: Interactive image segmentation with click imitation. In *European Conference on Computer Vision*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *SIGDIAL Conference*.

Ma, J., He, Y., Li, F., Han, L.-J., You, C., and Wang, B. (2023). Segment anything in medical images. *Nature Communications*, 15.

Maier-Hein, L., Reinke, A., Christodoulou, E., Glocker, B., Godau, P., Isensee, F., Kleesiek, J., Kozubek, M., Reyes, M., Riegler, M., Wiesenfarth, M., Baumgartner, M., Eisenmann, M., Heckmann-Notzel, D., Kavur, A. E., Radsch, T., Tizabi, M. D., Ación, L., Antonelli, M., Arbel, T., Bakas, S., Bankhead, P., Benis, A., Cardoso, M. J., Cheplygina, V., Cimini, B. A., Collins, G. S., Farahani, K., van Ginneken, B., Hashimoto, D. A., Hoffman, M. M., Huisman, M., Jannin, P., Kahn, C. E., Karargyris, A., Karthikesalingam, A., Kenngott, H., Kopp-Schneider, A., Kreshuk, A., Kurç, T. M., Landman, B. A., Litjens, G. J. S., Madani, A., Maier-Hein, K. H., Martel, A. L., Mattson, P., Meijering, E. H. W., Menze, B. H., Moher, D., Moons, K. G. M., Muller, H., Nickel, F., Nichyporuk, B., Petersen, J., Rajpoot, N. M., Rieke, N., Saez-Rodriguez, J., Guti'errez, C. S., Shetty, S., van Smeden, M., Sudre, C. H., Summers, R. M., Taha, A. A., Tsaftaris, S. A., Calster, B. V., Varoquaux, G., and Jager, P. F. (2022). Metrics reloaded: recommendations for image analysis validation. *Nature Methods*, 21:195 – 212.

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

Mazaré, P.-E., Humeau, S., Raison, M., and Bordes, A. (2018a). Training millions of personalized dialogue agents. In *EMNLP*.

Mazaré, P.-E., Humeau, S., Raison, M., and Bordes, A. (2018b). Training millions of personalized dialogue agents. In *Conference on Empirical Methods in Natural Language Processing*.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *ICLR*.

Mohapatra, S., Gosai, A., and Schlaug, G. (2023). Sam vs bet: A comparative study for brain extraction and segmentation of magnetic resonance images using deep learning. *ArXiv*, abs/2304.04738.

Monteiro, M., Folgoc, L. L., de Castro, D. C., Pawlowski, N., Marques, B., Kamnitsas, K., van der Wilk, M., and Glocker, B. (2020). Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty. *ArXiv*, abs/2006.06015.

Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., and Jin, Z. (2016). Natural language inference by tree-based convolution and heuristic matching. *arXiv: Computation and Language*.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2022). Mteb: Massive text embedding benchmark. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Mysore, S., Lu, Z., Wan, M., Yang, L., Menezes, S., Baghaee, T., Gonzalez, E. B., Neville, J., and Safavi, T. (2023). Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. *ArXiv*, abs/2311.09180.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *ICML*.

Ouguz, B., Lakhotia, K., Gupta, A., Lewis, P., Karpukhin, V., Piktus, A., Chen, X., Riedel, S., tau Yih, W., Gupta, S., and Mehdad, Y. (2021). Domain-matched pre-training tasks for dense retrieval. *ArXiv*, abs/2107.13602.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L. E., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. J. (2022). Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Oğuz, B., Lakhotia, K., Gupta, A., Lewis, P., Karpukhin, V., Piktus, A., Chen, X., Riedel, S., tau Yih, W., Gupta, S., and Mehdad, Y. (2021). Domain-matched pre-training tasks for dense retrieval. *ArXiv*, abs/2107.13602.

Palin, K., Feit, A. M., Kim, S., Kristensson, P. O., and Oulasvirta, A. (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*.

Pan, Y., Li, N., Gao, C., Chang, J., Niu, Y., Song, Y., Jin, D., and Li, Y. (2023). Learning and optimization of implicit negative feedback for industrial short-video recommender system. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*.

Pang, R. Y. and He, H. (2020). Text generation by learning from demonstrations. *ArXiv*, abs/2009.07839.

Paranjape, A., Khattab, O., Potts, C., Zaharia, M., and Manning, C. D. (2022). Hindsight: Posterior-guided training of retrievers for improved open-ended generation. In *International Conference on Learning Representations*.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 488–489.

Pennington, J., Socher, R., and Manning, C. (2014a). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014b). Glove: Global vectors for word representation. In *EMNLP*.

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., and Hinton, G. E. (2017). Regularizing neural networks by penalizing confident output distributions. *ArXiv*, abs/1701.06548.

Price, I., Gifford-Moore, J., Flemming, J., Musker, S., Roichman, M., Sylvain, G., Thain, N., Dixon, L., and Sorensen, J. S. (2020). Six attributes of unhealthy conversations. In *Workshop on Abusive Language Online*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019a). Language models are unsupervised multitask learners.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019b). Language models are unsupervised multitask learners.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Rahimi, S., Oktay, O., Alvarez-Valle, J., and Bharadwaj, S. (2021). Addressing the exorbitant cost of labeling medical images with active learning. In *International Conference on Machine Learning in Medical Imaging and Analysis*. Best Presentation Award.

Rahman, A., Valanarasu, J. M. J., Hacihaliloglu, I., and Patel, V. (2023). Ambiguous medical image segmentation using diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11536–11546.

Ramos, J. E. (2003). Using tf-idf to determine word relevance in document queries.

Rastogi, A., Zang, X., Sunkara, S., Gupta, R., and Khaitan, P. (2020). Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

Richardson, C., Zhang, Y., Gillespie, K., Kar, S., Singh, A., Raeesy, Z., Khan, O. Z., and Sethy, A. (2023). Integrating summarization and retrieval for enhanced personalization via large language models. *ArXiv*, abs/2310.20081.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Shuster, K., Smith, E. M., Boureau, Y.-L., and Weston, J. (2020). Recipes for building an open-domain chatbot. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E. M., Boureau, Y.-L., and Weston, J. (2021). Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2021). High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685.

Roy, S., Wald, T., Koehler, G., Rokuss, M. R., Disch, N., Holzschuh, J., Zimmerer, D., and Maier-Hein, K. H. (2023). Sam.md: Zero-shot medical image segmentation capabilities of the segment anything model. *ArXiv*, abs/2304.05396.

Rupprecht, C., Laina, I., DiPietro, R., Baust, M., Tombari, F., Navab, N., and Hager, G. D. (2016). Learning in an uncertain world: Representing ambiguity through multiple hypotheses.

Salemi, A., Kallumadi, S., and Zamani, H. (2024). Optimization methods for personalizing large language models through retrieval augmentation. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Salemi, A., Mysore, S., Bendersky, M., and Zamani, H. (2023). Lamp: When large language models meet personalization. *ArXiv*, abs/2304.11406.

Sanchez, G., Fan, H., Spangher, A., Levi, E., Ammanamanchi, P. S., and Biderman, S. (2023). Stay on topic with classifier-free guidance. *ArXiv*, abs/2306.17806.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019a). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019b). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., and Zaharia, M. A. (2021). Colbertv2: Effective and efficient retrieval via lightweight late interaction. *ArXiv*, abs/2112.01488.

Santos, F., Treviso, M., Gama, S., and Fortes, R. (2022). *A Framework to Semi-automated Usability Evaluations Processing Considering Users' Emotional Aspects*, pages 419–438.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *ArXiv*, abs/2302.04761.

Schick, T. and Schütze, H. (2021a). Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Schick, T. and Schütze, H. (2021b). Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T. P., and Silver, D. (2019). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588:604 – 609.

Shuster, K., Poff, S., Chen, M., Kiela, D., and Weston, J. (2021a). Retrieval augmentation reduces hallucination in conversation. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shuster, K., Urbanek, J., Szlam, A., and Weston, J. (2021b). Am i me or you? state-of-the-art dialogue models cannot maintain an identity. In *NAACL-HLT*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T. P., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T. P., Simonyan, K., and Hassabis, D. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *ArXiv*, abs/1712.01815.

Singh, I., Singh, G., and Modi, A. (2022). Pre-trained language models as prior knowledge for playing text-based games. In *AAMAS*.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958.

Sukhbaatar, S., Szlam, A. D., Weston, J., and Fergus, R. (2015). End-to-end memory networks. In *NIPS*.

Sun, S., Zhao, W., Manjunatha, V., Jain, R., Morariu, V., Dernoncourt, F., Srinivasan, B. V., and Iyyer, M. (2021). IGA: An intent-guided authoring assistant. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5972–5985, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *ArXiv*, abs/1409.3215.

Sutton, R. S. and Barto, A. G. (2005). Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 16:285–286.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Tahami, A. V., Ghajar, K., and Shakery, A. (2020). Distilling knowledge for fast retrieval-based chat-bots. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Tao, C., Wu, W., Xu, C., Hu, W., Zhao, D., and Yan, R. (2019). One time of interaction may not be enough: Go deep with an interaction-over-interaction network for response selection in dialogues. In *ACL*.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Tay, Y., Tran, V. Q., Dehghani, M., Ni, J., Bahri, D., Mehta, H., Qin, Z., Hui, K., Zhao, Z., Gupta, J., Schuster, T., Cohen, W. W., and Metzler, D. (2022). Transformer memory as a differentiable search index. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Teevan, J., Dumais, S. T., and Liebling, D. J. (2008). To personalize or not to personalize: modeling queries with variation in user intent. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 163–170, New York, NY, USA. Association for Computing Machinery.

Tian, K., Xu, Y., Zhou, S., and Guan, J. (2019). Versatile multiple choice learning and its application to vision computing.

Towle, B. and Zhou, K. (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.

Towle, B. and Zhou, K. (2023a). End-to-end autoregressive retrieval via bootstrapping for smart reply systems. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7610–7622, Singapore. Association for Computational Linguistics.

Towle, B. and Zhou, K. (2023b). Model-based simulation for optimising smart reply. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12030–12043, Toronto, Canada. Association for Computational Linguistics.

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

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017a). Attention is all you need. *ArXiv*, abs/1706.03762.

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017b). Attention is all you need. In *Neural Information Processing Systems*.

Wang, C., Mao, S., Ge, T., Wu, W., Wang, X., Xia, Y., Tien, J., and Zhao, D. (2023a). Smart word suggestions for writing assistance. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11212–11225, Toronto, Canada. Association for Computational Linguistics.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. (2023b). Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E. H., Xia, F., Le, Q., and Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. (2020). Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Weng, Y., Zheng, H., Bell, F., and Tür, G. (2019). Occ: A smart reply system for efficient in-app communications. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Wolf, T., Sanh, V., Chaumond, J., and Delangue, C. (2019). Transfertransfo: A transfer learning approach for neural network based conversational agents. *ArXiv*, abs/1901.08149.

Wo'zniak, S., Koptyra, B., Janz, A., l aw Kazienko, P., and Koco'n, J. (2024). Personalized large language models. *ArXiv*, abs/2402.09269.

Wu, L. Y., Fisch, A., Chopra, S., Adams, K., Bordes, A., and Weston, J. (2018). Starspace: Embed all the things! *ArXiv*, abs/1709.03856.

Wu, Y., Wu, W. Y., Zhou, M., and Li, Z. (2017). Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. *ArXiv*, abs/1612.01627.

Xu, J., Ju, D., Li, M., Boureau, Y.-L., Weston, J., and Dinan, E. (2020). Recipes for safety in open-domain chatbots. *ArXiv*, abs/2010.07079.

Xue, G.-R., Han, J., Yu, Y., and Yang, Q. (2009). User language model for collaborative personalized search. *ACM Trans. Inf. Syst.*, 27(2).

Yang, K. and Klein, D. (2021). FUDGE: Controlled text generation with future discriminators. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Yao, S., Narasimhan, K., and Hausknecht, M. J. (2021). Reading and acting while blindfolded: The need for semantics in text game agents. In *NAACL*.

Yao, S., Rao, R., Hausknecht, M., and Narasimhan, K. (2020). Keep calm and explore: Language models for action generation in text-based games. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Young, S. J., Gasic, M., Thomson, B., and Williams, J. (2013). Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101:1160–1179.

Yuan, C., jie Zhou, W., Li, M., Lv, S., Zhu, F., Han, J., and Hu, S. (2019). Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In *EMNLP*.

Zhang, A., Goyal, A., Kong, W., Deng, H., Dong, A., Chang, Y., Gunter, C. A., and Han, J. (2015). adaqac: Adaptive query auto-completion via implicit negative feedback. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zhang, C., Wang, H., Jiang, F., and Yin, H. (2021a). Adapting to context-aware knowledge in natural conversation for multi-turn response selection. *Proceedings of the Web Conference 2021*.

Zhang, M., Wang, W., Deb, B., Zheng, G., Shokouhi, M., and Awadallah, A. H. (2021b). 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.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A. D., Kiela, D., and Weston, J. (2018a). Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A. D., Kiela, D., and Weston, J. (2018b). Personalizing dialogue agents: I have a dog, do you have pets too? In *Annual Meeting of the Association for Computational Linguistics*.

Zhang, X., Zhang, Y., Long, D., Xie, W., Dai, Z., Tang, J., Lin, H., Yang, B., Xie, P., Huang, F., Zhang, M., Li, W., and Zhang, M. (2024). mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *ArXiv*, abs/2407.19669.

Zhang, Y., Hare, J. S., and Prügel-Bennett, A. (2019a). Deep set prediction networks. *ArXiv*, abs/1906.06565.

Zhang, Y. and Ippolito, D. (2023). Effective prompt extraction from language models.

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, W. B. (2019b). Dialogpt : Large-scale generative pre-training for conversational response generation. In *Annual Meeting of the Association for Computational Linguistics*.

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, W. B. (2020). Dialogpt : Large-scale generative pre-training for conversational response generation. In *ACL*.

Zhao, T., Lee, K., and Eskénazi, M. (2018). Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *ACL*.

Zhao, T., Xie, K., and Eskénazi, M. (2019a). Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *ArXiv*, abs/1902.08858.

Zhao, T., Zhao, R., and Eskénazi, M. (2017a). Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*.

Zhao, T., Zhao, R., and Eskénazi, M. (2017b). Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*.

Zhao, X., Tao, C., Wu, W., Xu, C., Zhao, D., and Yan, R. (2019b). A document-grounded matching network for response selection in retrieval-based chatbots. *ArXiv*, abs/1906.04362.

Zhou, L., Gao, J., Li, D., and Shum, H. (2020). The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, Just Accepted:1–62.

Zhou, X., Li, L., Dong, D., Liu, Y., Chen, Y., Zhao, W. X., Yu, D., and Wu, H. (2018). Multi-turn response selection for chatbots with deep attention matching network. In *ACL*.

Zhu, Y., Nie, J., Zhou, K., Du, P., and Dou, Z. (2021). Content selection network for document-grounded retrieval-based chatbots. *ArXiv*, abs/2101.08426.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593.