Titre: Title:	Spending Capacity Where It Matters: Selection, Adaptation, Interaction, and Structure for Efficient Language Understanding and Generation
Auteur: Author:	Jonathan Pilault
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Pilault, J. (2025). Spending Capacity Where It Matters: Selection, Adaptation, Interaction, and Structure for Efficient Language Understanding and Generation [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/71064/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/71064/
Directeurs de recherche: Advisors:	Christopher J. Pal
Programme: Program:	génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Spending Capacity Where It Matters: Selection, Adaptation, Interaction, and Structure for Efficient Language Understanding and Generation

JONATHAN PILAULT
Département de Génie Informatique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie Informatique

Décembre 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Spending Capacity Where It Matters: Selection, Adaptation, Interaction, and Structure for Efficient Language Understanding and Generation

présentée par **Jonathan PILAULT**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Matthew KUSNER, président
Christopher PAL, membre et directeur de recherche
Siva REDDY, membre
Olga VECHTOMOVA, membre externe

ACKNOWLEDGEMENTS

I feel deeply fortunate to have pursued my PhD at Mila. The lab's culture, curious, generous, and relentlessly exploratory, shaped how I think about research and collaboration. I am profoundly grateful to the entire Mila community, including students, postdocs, faculty, engineers, and administrative staff, whose kindness and professionalism made this work possible and joyful.

Above all, I owe heartfelt thanks to my advisor, Chris Pal, for being the kind of mentor every student hopes to find. From our very first meeting, when I was only a prospective student, his enthusiasm, positivity, and optimism set the tone for everything that followed. He gave me room to explore, a steady hand when I needed direction, and unwavering support when experiments failed or ideas changed shape. I hope to carry forward his example of scientific rigor paired with generosity and good humor.

I also want to thank my collaborators and friends at Mila and beyond, whose feedback sharpened ideas and whose encouragement kept them alive. The informal reading groups, code reviews, and late-evening debugging sessions were as formative as any formal course. I am equally grateful to the broader research community, including reviewers, organizers, and authors, whose work and critiques pushed this thesis to be clearer and stronger.

Finally, I owe the largest debt to my family. Their patience, love, and steady confidence sustained me through the inevitable peaks and valleys of every project. Many of the ideas here were inspired by watching children (Éléna, Mila, Rosalie, Sophia, Armel and Thalia) learn and evolve, see how they generalize, adapt, and negotiate ambiguity with surprising efficiency. Those everyday observations reminded me that intelligence is practical, social, and playful, and have quietly informed the directions I chose to pursue and will continue to pursue into the future.

To everyone who shared time, expertise, and encouragement, thank you.

RÉSUMÉ

Cette thèse adopte une approche programmatique visant à amener les grands modèles de langage à consacrer leur capacité là où celle-ci est la plus utile. Nous étudions quatre axes, chacun répondant à une contrainte pratique moderne du traitement automatique des langues, et proposons des conceptions qui, mises ensemble, tracent un chemin cohérent allant de la structuration des entrées à la structuration architecturale.

Dans le Chapitre 2, pour le résumé de documents longs, nous plaçons en amont du générateur Transformer une étape de sélection légère et contrainte, montrant que le conditionnement sur une esquisse compacte favorise une abstraction véritable plutôt que la copie de surface, sur des évaluations de longs textes telles que arXiv, PubMed, Newsroom et BigPatent.

Dans le Chapitre 3, pour l'apprentissage multi-tâches économe en paramètres, nous figeons la majeure partie des poids du modèle et entraînons de petits modules d'adaptation conditionnés par la tâche, jumelée à une politique d'échantillonnage explicite qui priorise les tâches selon la taille du jeu de données et l'incertitude prédictive (pondérée par l'incertitude) à budget de poids entraînés contraints. Notre adaptateur hyperréseau multi-tâches réduit l'interférence tout en améliorant le transfert sur plusieurs jeux de référence de classification de texte et de compréhension linguistique.

Dans le Chapitre 4, nous traitons l'ambiguïté inhérente à la génération linguistique qui apparaît lorsque les requêtes des utilisateurs omettent un contexte crucial. L'absence d'informations propres à l'utilisateur ou à la tâche conduit les modèles à produire des sorties plausibles mais mal alignées, ce qui dégrade la qualité et provoque souvent des reprises coûteuses. Au moyen d'évaluations ciblées en génération interlingue, nous reconcevons la génération comme l'étape finale d'une brève interaction qui pose, avant le décodage, des questions ciblées pour révéler les préférences manquantes (degré de formalité, réalisation du genre, résolution des pronoms). Conditionner le modèle sur ces clarifications l'oriente vers une réponse mieux alignée dès le premier passage, dans une courte phase d'élucidation, ce qui réduit l'incertitude sans allonger la génération.

Dans le Chapitre 5, pour la modélisation linguistique à longue portée, nous présentons les Block-State Transformers, une couche hybride combinant un modèle en espace d'état, pour une propagation efficace à long horizon, et une attention par blocs, pour un mélange local sensible au contenu, offrant des compromis perplexité-mémoire favorables sur de longues fenêtres de contexte ainsi que de solides performances sur les jeux de tests Long Range Arena.

Au fil des chapitres, le fil conducteur consiste à clarifier l'essentiel avant le décodage ou la prédiction - en sélectionnant, en conditionnant, en interagissant ou en hybridant - afin de concentrer la capacité préentraînée sur les parties de chaque problème qui en ont le plus besoin. Le Chapitre 6 synthétise ces axes et expose les liens avec les travaux récents ainsi que les prolongements futurs des LLM efficaces.

ABSTRACT

This thesis takes a programmatic view of getting large language models to spend capacity where it matters. We study four settings each addressing a practical pressure in modern Natural Language Processing and report designs that, taken together, form a coherent path from input structuring to architectural structuring.

In Chapter 2, for long-document summarization we place a lightweight, constrained selection step in front of a Transformer generator, showing that conditioning on a compact sketch encourages genuine abstraction over surface copying across long text evaluations such as arXiv, PubMed, Newsroom, and BigPatent.

In Chapter 3, for parameter-efficient multi-task learning we keep most of the Transformer layers frozen and route task variation through small, task-conditioned adapter modules, together with an explicit sampling policy that prioritizes tasks by dataset size (temperature-scaled) and predictive uncertainty (uncertainty-weighted) under a fixed trained-parameter budget. Our multi-task hypernetwork adapter reduces interference while improving transfer on several text classification and language understanding benchmarks with minimal added capacity.

In Chapter 4, we address the inherent ambiguity in language generation that arises when user queries omit crucial context. Missing user-specific or task-specific information leads models to produce plausible yet misaligned outputs, which degrades quality and often triggers costly retries or longer prompts. With specific evaluations on cross-lingual generation, we recast generation as the final step of a brief interaction that asks targeted pre-decoding questions to surface the missing preferences (formality, gender realization, pronoun resolution). Conditioning on these clarifications steers the model toward the intended output on the first pass, within a brief elicitation phase, reducing uncertainty without lengthening generation.

In Chapter 5, for long-range language modeling we introduce Block-State Transformers, a hybrid layer that composes a state-space models for efficient long-horizon propagation with block-wise attention for local, content-aware mixing, yielding favorable perplexity–memory trade-offs at long context and strong performance on Long Range Arena benchmarks.

Across chapters, the unifying theme is to clarify what matters before decoding or prediction by selecting, conditioning, interacting, or hybridizing so that pretrained capacity is focused on the parts of each problem that need it most. Chapter 6 synthesizes these threads and outlines links to current works and future extensions of efficient LLMs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Building computer systems that handle human language without brittle, hand-written rules has been a recurring ambition in AI. What began as carefully engineered pipelines gradually gave way to *learning-based* approaches that exploit the regularities latent in large text corpora. Early neural language models showed that next-word prediction could induce useful internal representations [41] where distributed word embeddings captured semantic and syntactic affinities by predicting a word from its context and vice versa [42]. Yet recurrent architectures effective at modeling sequences faced difficulties with long-range dependencies and limited parallelism, constraining both the breadth of phenomena they could capture and the practical scales at which they could be trained.

A decisive turn came with the Transformer, which replaces recurrence with self-attention [43]. Rather than propagating a state step by step, attention mixes information across positions in a single, highly parallel operation. Multi-head variants compute several such mixtures in parallel and then concatenate the results, allowing the model to represent a spectrum of dependency patterns. This design unlocked deeper networks, better hardware efficiency, and a shift toward pretraining on web-scale corpora. In encoder–decoder form (e.g., T5) the same mechanism supports conditional generation, while in decoder-only form (e.g., GPT-3) it enables left-to-right modeling with causal masks [44, 45].

Two pretraining paradigms became especially influential. The first is *autoregressive* modeling, in which a decoder-only stack learns to predict the next token given its history, which aligns directly with controlled generation and has proven surprisingly adaptable, with sufficiently large models exhibiting in-context learning and few-shot generalization [45]. The second paradigm is *masked language modeling* (MLM), in which an encoder reconstructs randomly masked tokens from bidirectional context. The resulting contextual representations transfer broadly after task-specific fine-tuning [46]. Unified text-to-text formulations subsequently framed diverse NLP tasks within a single interface, streamlining transfer and comparison [44]. Formal definitions and notation for these objectives and mechanisms are deferred to the *Definitions & base concepts* section.

As capabilities scaled, the field's attention shifted from *can the model do it?* to *does the model do what users actually want?* Instruction tuning and reinforcement learning from human feedback align pretrained models with user preferences without discarding their broad competencies [47]. At the same time, practical constraints model size, serving cost, and the need to support many tasks favored *parameter-efficient* adaptation. Rather than duplicating billions of parameters for every task, lightweight adapters, prefix/prompt tuning, bias-only updates, and low-rank modifications let practitioners keep a shared backbone and specialize behavior with a small trainable budget [48, 49, 50, 51, 52]. These developments set the stage for a thesis that treats large models not as monoliths to be fine-tuned wholesale, but as flexible substrates whose inputs, parameter pathways, and training protocols can be shaped to the demands of specific problems.

Against this backdrop, several recurring issues become salient in real applications. Long documents scientific articles, patents, legal filings routinely exceed typical context lengths. Even when longer contexts are feasible, naive conditioning can encourage extractive copying rather than faithful abstraction. The core difficulty is not only one of sequence length but also of *content selection*: deciding what to preserve, compress, and synthesize. Attention mechanisms spread capacity broadly. Without a mechanism to privilege salient fragments, generation can drift toward surface overlap instead of genuine distillation. A natural response is to introduce a lightweight selection step that filters the source before generation. Such a step narrows the model's field of view to what matters most, making the downstream decoder's job better posed.

A second issue arises when one model must serve many tasks across domains and data regimes. Full fine-tuning for each task is expensive to store and maintain, and naive joint training can lead to interference and forgetting. Parameter-efficient modules and careful multi-task sampling offer a more tractable path: keep much of the pretrained backbone fixed, and let small, task-conditioned components absorb the necessary specialization. This approach retains the benefits of pretraining while reducing the per-task footprint, enabling frequent iteration and deployment in resource-constrained settings. It also reframes "transfer" as a design space: which parameters should remain shared, which should be adapted, and how should data from heterogeneous tasks be mixed so that low-resource tasks are not drowned out?

A third issue concerns conditional generation under ambiguity, with machine translation as a canonical example. For many inputs there is not a single correct output but a family of acceptable choices whose selection depends on purpose, register, or domain conventions. A model that is fluent but misaligned with these latent preferences may produce outputs that are formally correct yet pragmatically wrong. Prompt hints can help, but short, lightweight interaction before decoding can be more reliable: by asking a few targeted questions, the system can resolve ambiguities and condition generation on explicit preferences. In effect, the model reduces uncertainty about the user's intent before committing to a single output, much as a careful human translator would.

Finally, there is the question of long-range modeling under runtime constraints. Self-attention's quadratic complexity in sequence length strains memory and latency for very long inputs. Efficient Transformer variants reduce this cost through sparsity, kernels, or linear approximations of attention while state-space models (SSMs) offer subquadratic recurrences with strong performance on certain long-sequence tasks [53]. Each family has complementary strengths: attention excels at capturing local, content-dependent interactions. SSMs provide scalable, stable mechanisms for long-range propagation. Composing them within a single layer promises a better quality–efficiency trade-off than either alone, especially when implemented to exploit model and data parallelism.

This is a thesis by article based on the following articles:

1. Pilault, J., Li, R., Subramanian, S. and Pal, C. (EMNLP 2020). On Extractive and Abstractive Neural Document Summarization with Transformer Language Models [54].

2. Pilault, J., Amine El hattami and Pal, C. (ICLR 2021). Conditionally Adaptive Multi-Task Learning: Improving Transfer Learning in NLP Using Fewer Parameters & Less Data [50].

3. Pilault, J., Fathi, M., Firat, O., Pal, C., Bacon, P.-L. and Goroshin, R. (NeurIPS 2023). Block-State Transformers [55].

4. Pilault, J., Garcia, X., Bražinskas, A. and Firat, O. (ICML 2023 workshop and IJCNLP-AACL 2023). Interactive-Chain-Prompting: Ambiguity Resolution for Crosslingual Conditional Generation with Interaction [56].

Together they frame the rest of the thesis, with each article treating one of the issues above and informing the design choices that follow. A common thread through in this work is to structure the problem so that model capacity is spent where it matters. In one setting, that means placing a small, extractive lens in front of a generator so that "abstractive" summarization need not ingest or process the entire document at once. In another, it means splitting "what is shared" from "what is adapted," and using principled sampling so that each task receives an appropriate gradient budget. In a third, it means treating generation as the last step in a short dialogue that clarifies the target before producing it. And in the last, it means building a hybrid layer that lets a state-space component carry long-range context while a block-wise attention component handles fine-grained, local structure.

The result is a program that moves from input structuring to parameter structuring to interaction structuring to architectural structuring. None of these choices is proposed as a universal fix since each is presented where it fits naturally. The ideas are intentionally modest in isolation simple selection in front of a Transformer, small adapter modules on top of a frozen backbone, a handful of targeted questions before decoding, a hybridization of two well-studied sequence mechanisms but together they illustrate a way of thinking about large models that is pragmatic and extensible. Start from the capabilities that pretraining affords, then add just enough structure at the input, in the parameters, in the protocol, or in the layer to make those capabilities usable under the constraints of the task.

The discussion proceeds in a sequence that mirrors this logic. We begin with long-document summarization, where a selective conditioning step is used to guide a Transformer toward genuine abstraction rather than surface copying. We then turn to multi-task learning, adopting task-conditioned modules and data sampling to improve transfer while freezing a substantial portion of the pretrained backbone. Next, we consider cross-lingual generation under ambiguity and show how a short elicitation phase can reduce uncertainty before decoding. Finally, we study long-range language modeling with a hybrid layer that composes state-space recurrences with block-wise attention to improve efficiency at extended sequence lengths. The concluding chapter draws connections among these strands and considers how the same organizing principles might extend to adjacent problems beyond those treated here.

1.1 Definitions & Base Concepts

This section gathers the background that a technically trained reader needs in order to follow the remainder of this thesis without consulting external primers. The emphasis is on *definitions* and *canonical formulations* rather than on any particular research agenda. Here, we fix terms, summarize standard objectives, and describe widely used architectural and evaluation primitives in contemporary natural language processing (NLP) with large language models (LLMs).

We proceed from the ground up: (1) sequences and tokenization in Section 1.1.1; (2) the Transformer abstraction in Section 1.1.2; (3) pretraining objectives and alignment in Section 1.1.3; (4) decoding in Section 1.1.4; (5) evaluation signals commonly encountered in summarization, translation, and language modeling in Section 1.1.5; (6) parameter-efficient adaptation (PEFT) in Section 1.1.6; (7) multi-task training primitives in Section 1.1.7; (8) ambiguity and user-conditioned generation in Section 1.1.8; (9) long-sequence modeling strategies including efficient attention and state-space models (SSMs) in Section 1.1.9; (10) content selection and abstraction in Section 1.1.10; (11) data, preprocessing, and distributions in Section 1.1.11; and (12) training and optimization basics in Section 1.1.12; For convenience, we also summarize key definitions in Section 1.1.14. To keep the discussion modular, each concept is introduced independently of downstream methodology, and notation is kept light and consistent across subsections.

1.1.1 Sequences, tokenization, and embeddings

Discrete sequences. NLP models operate on sequences of discrete symbols drawn from a finite vocabulary. After normalization (e.g., lowercasing, punctuation handling) and segmentation, a text x is mapped to a token sequence $x_{1:L} = (x_1, \dots, x_L)$, where L is the sequence length and each $x_i \in \mathcal{V}$ indexes a symbol.

Subword segmentation. Because word-level vocabularies incur out-of-vocabulary (OOV) issues and morphological brittleness, modern pipelines rely on *subword* schemes. Two families dominate: byte-pair encoding (BPE) and unigram language-model tokenization. BPE iteratively merges frequent symbol pairs, trading vocabulary growth for shorter sequences; unigram tokenization learns a probabilistic inventory and chooses segmentations that maximize likelihood. Both approaches aim to (i) cap the vocabulary size, (ii) decompose rare words into common parts, and (iii) keep frequent words intact to preserve statistical efficiency.

Embeddings. Each token index x_i is mapped to a d-dimensional vector $e_i \in \mathbb{R}^d$ via a learned embedding table $E \in \mathbb{R}^{|\mathcal{V}| \times d}$. The embedding stage is the entry point for contextual modeling: it provides continuous inputs to layers that mix information across positions. Embedding tables may be *tied* to output classifiers (weight sharing) in autoregressive (AR) decoders to regularize training.

Segmenting long inputs. When documents exceed maximum context length, they are typically split into overlapping or non-overlapping windows, each of which is processed independently or with limited carry-over state. The choice of window length, stride, and ordering affects which dependencies are representable within a forward pass and what must be captured through recurrence, retrieval, or global tokens in the architecture (see Section 1.1.9).

1.1.2 Transformer fundamentals

Self-attention as content-based mixing. The Transformer eschews recurrence in favor of *self-attention*, a mechanism that computes, for each position, a weighted average of representations at all positions, where weights are content-dependent. Given an input matrix $\mathbf{X} \in \mathbb{R}^{L \times d_{\text{model}}}$, the layer forms queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ through learned linear projections and computes

$$\operatorname{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \;=\; \operatorname{softmax}\!\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{1.1}$$

where d_k is the key dimension. The softmax normalizes attention weights row-wise so they sum to one at each query position.

Multi-head attention and feed-forward sublayers. In practice, h attention "heads" operate in parallel on $d_k = d_{\text{model}}/h$-dimensional subspaces, their outputs are concatenated and linearly reprojected. Each Transformer block includes residual connections around attention and around a position-wise feed-forward network (FFN) of the form

$$\mathrm{FFN}(\mathbf{z}) \;=\; \phi(\mathbf{z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2,$$

with nonlinearity ϕ (e.g., GeLU), plus layer normalization. Modern stacks use pre-norm ordering (LayerNorm → sublayer → residual add) for stability at depth.

Positional information. Since attention is permutation-equivariant, models inject order through *positional encodings*. Absolute encodings (e.g., sinusoidal) add position-dependent vectors to embeddings and relative/rotary encodings incorporate position differences in attention score computation. The absolute sinusoidal encodings is defined in equation 1.2 below.

$$\begin{aligned}
&\forall\,\text{position } i \geq 0 \text{ and even channel index } 0 \leq 2k < d_{\text{model}},\\
&\mathrm{PE}(i, 2k) \;=\; \sin\!\left(i \,\Big/\, 10000^{\frac{2k}{d_{\text{model}}}}\right),\\
&\mathrm{PE}(i, 2k{+}1) \;=\; \cos\!\left(i \,\Big/\, 10000^{\frac{2k}{d_{\text{model}}}}\right),
\end{aligned} \tag{1.2}$$

where i is the token position, k is the channel pair index for sine/cosine, d_model is the model embedding dimension and $PE(i,$ ů$)$ is positional embedding vector at position i.

Absolute encodings are simple and robust but relative schemes often provide better extrapolation to lengths beyond those seen in training.

Masks and model families. Causal masks prevent a token from attending to future positions, enabling left-to-right AR modeling in decoder-only stacks. Encoder-only stacks (e.g., masked language models) use bidirectional attention without masks. Encoder–decoder stacks combine a bidirectional encoder with a causal decoder and add *cross-attention* that lets decoder queries attend to encoder outputs, a standard interface for sequence-to-sequence tasks.

Computational profile. While self-attention performs $O(L^2)$ score computations per head per layer, FFNs consume $O(Ld_{\text{model}}d_{\text{ff}})$ flops with d_{ff} the hidden size. For moderate L, FFNs dominate compute and, for large L, attention dominates both compute and memory due to the $L \times L$ score tensor. This profile motivates efficient attention designs (Section 1.1.9) when documents or contexts are long.

1.1.3 Pretraining objectives and adaptation

Autoregressive language modeling. AR models (e.g., GPT-family) optimize next-token prediction:

$$\mathcal{L}_{\mathrm{AR}}(\theta) \;=\; -\sum_{t=1}^{L} \log p_\theta(x_t \mid x_{<t}), \qquad p_\theta(x_t \mid x_{<t}) = \mathrm{softmax}\Big(\mathbf{h}_t \mathbf{W}_{\mathrm{vocab}}\Big)_{x_t}. \tag{1.3}$$

They directly support free-form and conditional generation by seeding the model with a prefix and sampling continuations. The objective is language-agnostic and scales cleanly with data, parameters, and context length [45].

Masked language modeling. Masked LMs (e.g., BERT) train encoders to reconstruct randomly masked tokens from bidirectional context:

$$\mathcal{L}_{\mathrm{MLM}}(\theta) \;=\; -\mathbb{E}_M\left[\sum_{i \in M} \log p_\theta(x_i \mid x_{\setminus M})\right], \tag{1.4}$$

with M a set of masked indices and $x_{\setminus M}$ the visible tokens. MLM tends to learn strong token-level representations that transfer well to classification and span prediction.

Denoising sequence-to-sequence. Encoder–decoder models (e.g., T5) use span corruption: contiguous spans are replaced with sentinels, and the target is the concatenation of missing spans. This unifies diverse tasks under a text-to-text interface [44].

Instruction tuning and human feedback. Alignment methods adapt pretrained LMs to follow natural-language instructions. *Instruction tuning* (a form of supervised fine-tuning) uses curated instruction–response pairs to teach prompt following. *Reinforcement learning from human feedback* (RLHF) adds a learned reward model (from pairwise human preferences) and policy optimization to shift generation toward preferred behaviors [47]. These techniques are *orthogonal* to pretraining objective choice: AR and seq2seq bases can both be instruction-tuned.

Catastrophic forgetting and regularization. Fine-tuning large models on narrow tasks can degrade previously acquired competencies. Regularization (e.g., weight decay, dropout), careful learning-rate schedules, and freezing large subsets of parameters mitigate this effect. Parameter-efficient methods (next subsection) further reduce the surface area susceptible to forgetting by constraining updates.

1.1.4 Decoding

Greedy and beam search. Greedy decoding chooses the highest-probability token at each step, cheap but prone to dull or repetitive outputs. Beam search maintains B candidate sequences, expanding each by top-scoring tokens and pruning to B. A *length penalty* $\ell(\cdot)$ can be applied to avoid short hypotheses. Further, repetition penalties and coverage penalties mitigate degeneracies in some tasks. Beam search remains standard for tasks with low-entropy targets (e.g., translation under strong references), though its utility decreases for open-ended generation where diversity matters.

Stochastic sampling. Temperature scaling divides logits by $\tau > 0$ where higher τ increases entropy. *Top-k sampling* restricts draws to the k most probable tokens. *Nucleus sampling* (top-p) samples from the smallest set of tokens whose cumulative probability exceeds p. Sampling is preferred when multiple valid verbalizations exist or when stylistic variety is beneficial.

Constrained decoding. For tasks with structural constraints (e.g., bracketed outputs, JSON, keyword inclusion), constrained decoders restrict available tokens or impose finite-state automata (FSA)-style masks over the vocabulary at each step. Constrained methods can also enforce lexicon constraints in translation and entity preservation in summarization, though they may interact with language modeling quality.

1.1.5 Evaluation signals

Overlap-based metrics. ROUGE-n recall measures n-gram overlap between a system output y and reference $y^\star$:

$$\text{ROUGE-}n \;=\; \frac{\sum_{g\in\mathcal{G}_n(y^\star)} \min\Big(\text{cnt}(g,y), \text{cnt}(g,y^\star)\Big)}{\sum_{g\in\mathcal{G}_n(y^\star)} \text{cnt}(g,y^\star)}. \tag{1.5}$$

BLEU computes a brevity-penalized geometric mean of modified n-gram precisions over $n \leq 4$. Overlap metrics are inexpensive and allow system-level comparisons across corpora, but they reward surface similarity and penalize legitimate paraphrase, especially when references are sparse or short.

Learned metrics. Reference-based learned evaluators (e.g., BLEURT, COMET) regress from system output(s) and reference(s) to human judgment targets. They offer improved correlation with adequacy and meaning preservation but depend on the diversity and domain coverage of their training data.

Language-modeling metrics. Perplexity (PPL) summarizes the predictive fit of an AR model on a tokenized corpus:

$$\mathrm{PPL} \;=\; \exp\!\left(\frac{-1}{\sum_i |x^{(i)}|}\sum_i \sum_{t=1}^{|x^{(i)}|} \log p_\theta(x_t^{(i)} \mid x_{<t}^{(i)})\right). \tag{1.6}$$

PPL is sensitive to tokenization and domain mismatch and is not a guarantee of downstream task performance, yet, it is best interpreted as a pretraining or modeling diagnostic.

Human evaluation. Quantitative human studies typically rate *fluency*, *coherence*, *informativeness*, and *factuality* on Likert scales or via pairwise preferences. Measurement quality depends on rater expertise, instructions, and inter-annotator agreement. For tasks involving multiple valid outputs (e.g., open-ended summarization, style-sensitive translation), calibrated human studies complement automatic metrics by capturing qualities that lexical overlap cannot.

Copying and abstractiveness. To characterize the nature of generated summaries, one can measure the fraction of n-grams in the output that also appear in the source (copy ratio) and its complement (novelty). While purely extractive methods may score well on overlap metrics, they tend to exhibit high copy ratios. On the other hand, more abstractive models generate novel n-grams and require different evaluation baselines.

1.1.6 Parameter-efficient fine-tuning (PEFT)

Motivation. Full fine-tuning of a large LM for each task or client implies duplicating all parameters and optimizer states, which is often costly in storage and compute, and can increase the risk of catastrophic forgetting. PEFT approaches freeze most of the backbone and train small, task-specific modules or low-rank updates.

Adapters. Adapters insert bottleneck MLPs inside each Transformer block and update only the adapter parameters while freezing the rest [48]. A typical adapter computes

$$\mathrm{Adapter}(\mathbf{h}) \;=\; \mathbf{h} + \mathbf{W}_{\mathrm{up}}\,\phi\!\Big(\mathbf{W}_{\mathrm{down}}\mathbf{h}\Big),$$

with $\mathbf{W}_{\mathrm{down}} \in \mathbb{R}^{r\times d}$, $\mathbf{W}_{\mathrm{up}} \in \mathbb{R}^{d\times r}$, and $r \ll d$. Variants differ in placement (after attention, after FFN, or both), nonlinearity, and whether residual scaling is learned. *AdapterFusion* learns to combine multiple trained adapters, offering a way to reuse prior task expertise [57].

BitFit. Bias-only tuning updates just the bias terms throughout the network [58]. Despite its simplicity and tiny parameter count, BitFit can capture a non-trivial fraction of the gains of full fine-tuning in classification settings, making it a baseline or a quick adaptation method.

LoRA (low-rank adaptation). LoRA attaches low-rank matrices to selected weight matrices (often attention projections) and learns those while freezing the base weights [59]. If $\mathbf{W}$ is frozen, LoRA learns $\Delta\mathbf{W} = \mathbf{BA}$ with $\mathbf{A} \in \mathbb{R}^{r\times d_{\text{in}}}$ and $\mathbf{B} \in \mathbb{R}^{d_{\text{out}}\times r}$, and uses $\mathbf{W}' = \mathbf{W}+\alpha\Delta\mathbf{W}$ at inference. The rank r controls the adaptation budget.

Prefix and prompt tuning. Prefix tuning learns a short sequence of *virtual* key–value pairs per layer that are concatenated to the actual keys and values, effectively providing a *task prefix* in attention space [49]. Prompt tuning instead learns continuous embeddings for a textual prefix at the input layer. Both keep the backbone frozen and are well-suited for generation tasks.

Implementation considerations. PEFT modules are composable: one may combine adapters with LoRA or with prefixes. They are also *swappable*: different tasks or clients load their own small modules on a shared backbone. Practical concerns include (i) parameter placement (which layers benefit most), (ii) interference when stacking multiple modules, (iii) initialization and rank selection for LoRA, and (iv) training stability when only a small part of the network updates.

1.1.7 Multi-task training primitives

Task sets and sampling. In multi-task learning (MTL), a model is trained over a set of tasks $\mathcal{T} = \{t\}$ with datasets $\{D_t\}$. Each training step samples a task according to a distribution $q(t)$ and updates parameters using a batch from D_t, the dataset for task t. We define the temperature-scaled size-based sampler as:

$$q_\alpha(t) \;=\; \frac{|D_t|^\alpha}{\sum_{s=1}^{T} |D_s|^\alpha}, \qquad \alpha \in [0,1], \tag{1.7}$$

where α is temperature that interpolates between uniform ($\alpha = 0$) and size-proportional ($\alpha = 1$) and $q_\alpha(t)$ is the probability of sampling task t. Common sampling policies include (i) *proportional* to dataset size, (ii) *temperature-scaled* $q(t) \propto |D_t|^\alpha$ with $\alpha \in [0,1]$ to upweight low-resource tasks, and (iii) *uncertainty-based* in equation 1.8 below, which draws tasks or examples with higher predictive entropy more often:

$$\begin{aligned} H_t \;&=\; \mathbb{E}_{(x,y)\sim\mathcal{B}_t}\Big[\,H\Big(p_\theta(\cdot \mid x,t)\Big)\,\Big], \\ q_{\text{unc}}(t) \;&=\; \frac{H_t^\beta}{\sum_{s=1}^{T} H_s^\beta}, \qquad \beta > 0, \end{aligned} \tag{1.8}$$

where $\mathcal{B}_t$ is a small probe batch drawn from D_t, $p_\theta(\cdot \mid x, t)$ is a model predictive distribution for task t on input x, $H(\cdot)$ is the Shannon entropy, H_t is average predictive uncertainty on task t, β is the concentration parameter that emphasizes higher-uncertainty tasks and $q_{\text{unc}}(t)$ is the probability of sampling task t by chance. The choice of q affects gradient magnitudes and forgetting dynamics.

Conditioning inductive biases. Multi-task setups often supply a *task embedding* or task identifier (e.g., a learned vector z_t) that conditions adapters, prefixes, or attention biases. Conditioning can be global (single vector used throughout) or local (different signals for different layers or heads). When task boundaries are fuzzy, *input-derived* signals (e.g., domain classifiers) can substitute for explicit task IDs. We will investigate in later chapters three main types of conditional inductive biases. The first is task-conditioned weight modulation defined in equation 1.9:

$$\phi(W \mid z_t) \;=\; \gamma(\mathbf{z_t}) \odot \mathbf{W} \;+\; \beta(\mathbf{z_t}), \tag{1.9}$$

where W is the base weight tensor of a layer (shape depends on the layer), $z_t \in \mathbb{R}^{d_z}$ is the embedding that encodes task/condition t, $\gamma(\cdot), \beta(\cdot)$ are small conditioning networks (e.g., MLPs) producing scale and shift, $\odot$ is an elementwise (Hadamard) product and $\phi(W \mid z_t)$ is the modulated weight tensor used at inference/training for task t. The second is a task-conditioned layer normalization defined in equation 1.10:

$$\mathrm{CLN}(h \mid z_t) \;=\; \frac{h - \mu(h)}{\sqrt{\sigma^2(h) + \epsilon}} \odot \gamma(\mathbf{z_t}) \;+\; \beta(\mathbf{z_t}), \tag{1.10}$$

where $h \in \mathbb{R}^d$ is a pre-activation vector, $\mu(h), \sigma^2(h)$ are per-feature mean and variance computed over the d features, ϵ is a small constant for numerical stability, $\gamma(z_t), \beta(z_t)$ are condition-dependent scale and shift produced from z_t and CLN is layer normalization whose affine parameters depend on the condition/task. The third is a task-conditioned attention bias defined in equation 1.11:

$$\mathbf{A}'_{ij} \;=\; \mathbf{A}_{ij} \;+\; u(\mathbf{z_t})^\top \mathbf{v}_{ij}, \tag{1.11}$$

where A_{ij} is the attention logit between query position i and key position j before bias, $u(z_t) \in \mathbb{R}^{d_b}$ is the bias vector produced from the condition embedding z_t, $v_{ij} \in \mathbb{R}^{d_b}$ is the feature vector encoding head/layer or relative-position metadata and A'_{ij} is the biased attention logit incorporating the condition/task signal.

Partial freezing. Beyond PEFT, one can freeze a subset of backbone layers while fine-tuning others (e.g., unfreeze only top k layers). This reduces compute and storage, constrains degrees of freedom, and can stabilize MTL by limiting the scope of interference.

1.1.8 Ambiguity and user-conditioned generation

Ambiguity types. Natural language exhibits multiple ambiguity sources relevant to generation: lexical (polysemy), structural (attachment, anaphora), pragmatic (formality, register), and socio-linguistic (gendered forms, honorifics). In cross-lingual settings, grammatical categories may be obligatory in the target language but underspecified in the source, requiring the model to pick among options (e.g., formality levels or gendered pronouns) in the absence of disambiguating context.

Preference elicitation. To reduce output variance stemming from latent preferences, models can be conditioned on small sets of *clarifications* collected prior to decoding. A practical pattern is an *interaction chain*: a short sequence of targeted questions about the intended style, domain, or referent properties, followed by answers that are then incorporated into the decoding prompt or hidden state. The chain makes implicit constraints explicit before decoding, which tightens the target distribution and steadies downstream choices.

Prompting vs. interaction. Static prompts (templates, instructions) provide indirect control and rely on the model to infer preferences from instructions embedded in a single input. In contrast, interactive approaches separate *preference discovery* (ask minimal questions that disambiguate) from *realization* (generate using those answers). Interaction is most helpful when a small number of answers resolve many downstream choices. In other words, interaction is less needed when the source already fixes the relevant attributes.

Evaluation considerations. For ambiguity-sensitive tasks, reference-based metrics may blur improvements if references reflect a different set of choices than the system output. Learned metrics and targeted test sets (e.g., pronoun resolution challenge sets) complement standard scores by measuring correctness along dimensions that hinge on preferences or world knowledge.

1.1.9 Long-sequence modeling

Why length matters? Many real-world inputs exceed a few thousand tokens: research articles, patents, legal filings, code repositories, minutes of meetings, or conversational logs. Representing dependencies at such lengths stresses memory and compute. The key challenge is to *separate concerns*: provide mechanisms for (i) local, high-bandwidth composition (word order, short-range syntax), (ii) mid-range structure (sentential and paragraph coherence), and (iii) global discourse signals (topic, coreference across sections) without paying quadratic costs everywhere.

Sparse attention patterns. A straightforward approach constrains attention to *local windows* of size W around each position, reducing complexity to $O(LW)$. Global tokens (learned or designated) can be added to allow any position to attend to a small set of summaries, enabling information to flow across windows at low cost. *Dilated* or *strided* patterns extend receptive fields without densely connecting all tokens. The design space includes blockwise attention (first split sequence into blocks then attend within block and to block summaries), landmark or routing tokens, and mixtures of local and global heads.

Kernelized/linear attention. Another route approximates the softmax kernel $\exp(\mathbf{q}^\top \mathbf{k})$ with a feature map $\phi(\cdot)$ so that

$$\mathrm{softmax}(\mathbf{Q}\mathbf{K}^\top)\mathbf{V} \;\approx\; \frac{\phi(\mathbf{Q})\Big(\phi(\mathbf{K})^\top \mathbf{V}\Big)}{\phi(\mathbf{Q})\Big(\phi(\mathbf{K})^\top \mathbf{1}\Big)},$$

enabling associativity to compute in $O(Ld)$ time and memory. Choices of ϕ (random features, positive-definite kernels) determine approximation quality and stability. Linear attention supports streaming but can degrade on tasks that rely on exact softmax behavior.

Memory tokens and recurrent summaries. Some architectures maintain a small bank of *memory tokens* that accumulate information across segments. After each segment is processed, a summary is written into the memory bank, such that:

$$H_k \;=\; \mathrm{Transformer}\Big(\,[M_{k-1};\, S_k]\,\Big), \tag{1.12}$$

where the long sequence is partitioned into segments $S_1, \ldots, S_K$., S_k is the token matrix for segment k (length L_k by d_{model}), $M_{k-1} \in \mathbb{R}^{m \times d_{\mathrm{model}}}$ is the memory tokens carried from segment $k-1$, $[\,\cdot\,;\,\cdot\,]$ is the concatenation along the sequence length dimension and H_k are the hidden states output for segment k (including positions for memory and S_k). Afterwards, the next segment attends to those memories alongside the current tokens. This trades a small global attention cost for a large reduction in pairwise token interactions.

Retrieval-augmented modeling. Rather than storing facts in parameters or attending over huge contexts, retrieval-augmented models fetch relevant passages from an external index at inference time and condition generation on them. Retrieval can be (i) *dense* (neural embeddings) or (ii) *sparse* (BM25-like), and (iii) *one-shot* (once per query) or (iv) *iterative* (interleaved with generation). The key benefit is decoupling world knowledge storage from parametric capacity. The key risk is retrieval errors or drift when indices are not refreshed.

State-space models (SSMs). SSMs model sequences using linear time-invariant dynamics with input-dependent driving and output readout. In discrete time, with input u_t, hidden state h_t, and output y_t,

$$h_{t+1} = Ah_t + Bu_t, \qquad y_t = Ch_t + Du_t.$$

Unrolling yields a 1D convolution $y = K * u$ with kernel $K = (CB, CAB, \dots, CA^{L-1}B)$. FFT-based implementations can compute this efficiently for long L. Structured choices of A and C (e.g., diagonal-plus-low-rank or normal plus low-rank) improve stability and expressivity in long-horizon regimes [60]. while SSMs are less naturally content-addressable than attention, SSMs provide (1) linear-time inference, (2) constant memory in sequence length, and (3) good inductive bias for certain modalities.

Hybrids of attention and SSMs. Hybrid designs combine attention's strength at content lookup with SSMs' strength at long-range propagation. Composition strategies include (1) stacking (alternating blocks), (2) parallel branches with learned fusion, and (3) blockwise hybrids in which attention operates locally while an SSM branch carries global context. Hybrids seek to approximate the quality of full attention at lower cost by assigning different parts of the modeling job to different mechanisms.

Engineering strategies. Regardless of architecture, scaling to long sequences benefits from careful engineering: (1) mixed-precision training to reduce memory [61], (2) activation checkpointing or rematerialization to trade compute for memory, (3) fused attention kernels [62], (4) gradient accumulation to maintain effective batch sizes [63], and (5) memory-conserving optimizer states [64]. These techniques do not change model semantics but determine feasible sequence lengths and batch sizes.

Later, we use a hybrid that makes this split explicit: SSMs carry long-horizon signals and block-attention handles local mixing.

1.1.10 Content selection and abstraction

Extractive vs. abstractive summaries. In *extractive* summarization, the output is assembled from spans (typically sentences) copied from the source. In *abstractive* summarization, the output is generated freely and may paraphrase, compress, or reorder information. While extractive methods often excel in faithfulness and ROUGE recall, abstractive methods can produce more concise and fluent text but risk factual errors if the modeling or conditioning signal is weak.

Selection heuristics and neural extractors. For long inputs, a common strategy is to build a compact *content sketch* by selecting top-k sentences. While heuristics (e.g., TF–IDF, position bias) provide simple baselines, neural extractors score sentences with contextual encoders and select those with highest scores, possibly under diversity constraints. Extractive stages can be trained using pseudo-labels derived from reference summaries (e.g., ROUGE-based matching). The purpose of selection is to improve the *conditioning* of a generator by focusing it on salient content.

Measuring abstractiveness. n-gram copy ratios quantify how much of the output overlaps with the source. Low copy ratios indicate more paraphrase and compression and high ratios indicate extraction. Neither extreme guarantees quality: too much novelty risks hallucination and too much copying undermines usefulness when concision and synthesis are required.

Factuality and faithfulness. Abstractive systems can produce *intrinsic* hallucinations (contradicting the source) or *extrinsic* hallucinations (introducing unsupported facts). Faithfulness can be probed with entailment-based checks or with human evaluation focused on information consistency. Conditioning signals that preserve provenance (e.g., sentence IDs) can facilitate auditing or post-hoc verification.

1.1.11 Data, preprocessing, and distributions

Domain and register. Data sources vary in domain (news, scientific text, conversational data) and register (formal vs. informal), which affects token distributions, sentence lengths, and discourse structure. Pretraining on mixed domains may improve robustness. Task-specific fine-tuning benefits from matching the target domain as closely as possible or from domain-adaptive pretraining.

Document structure. Long-form documents exhibit structure (titles, abstracts, sections, headings) that can be preserved as *segment tags* during preprocessing. Keeping lightweight structure (e.g., marking introductions or summaries) can help models learn customary content placement, even when the modeling architecture is unchanged.

Normalization and cleaning. Standard steps include Unicode normalization, de-duplication (to avoid train/test contamination or repetition), language detection, and removal of boilerplate. Tokenization choices (BPE vs. unigram LM) interact with normalization. For example, aggressive normalization may reduce the need for large vocabularies but can erase useful casing cues.

Sequence packing. For efficiency, multiple short sequences can be *packed* into a single long training example with separators and attention masks, improving accelerator utilization. Packing introduces cross-example attention unless masks are applied. When using causal masks, care is needed to prevent targets from attending to neighboring sequences.

1.1.12 Training and optimization basics

Optimizers and schedules. Adam [65] and its variants remain standard for training Transformers. Warmup schedules gradually increase learning rates from zero for the first few thousand steps before decaying linearly or with cosine schedules. Weight decay regularizes and gradient clipping stabilizes training in the face of rare large gradients.

Initialization and normalization. Proper initialization avoids signal blow-up or attenuation at depth. Pre-norm Transformers place LayerNorm before sublayers, improving gradient flow. Residual scaling (e.g., with learned gate parameters) can further stabilize very deep networks.

Regularization. Dropout applied to attention weights and FFN activations reduces overfitting. Label smoothing regularizes the target distribution in classification settings and may help in seq2seq training. Data augmentation in text (back-translation, noising) can increase robustness.

Mixed precision and numerical stability. Automatic mixed precision (AMP) speeds up training and reduces memory [61], but requires care with operations sensitive to underflow/overflow (e.g., softmax, layer norm). Loss scaling prevents gradient underflow in fp16. For example:

$$\begin{aligned} &\tilde{\mathcal{L}} \;=\; s\cdot\mathcal{L}, \qquad g \;=\; \nabla\tilde{\mathcal{L}}\Big/s, \\ &\text{on overflow: } s \leftarrow s/\kappa, \qquad \text{otherwise (periodically): } s \leftarrow s\cdot\kappa, \end{aligned} \tag{1.13}$$

where $\mathcal{L}$ is the original loss computed in mixed precision, $s > 0$ is the dynamic loss scale factor, $\tilde{\mathcal{L}}$ is the scaled loss used for backprop to avoid underflow, g is the unscaled gradients after dividing by s, $\kappa > 1$ is the multiplicative factor used to adjust the scale and "overflow" is the detection of inf/NaN in gradients or optimizer state.

1.1.13 Ethical and societal considerations

Bias and fairness. Language models may encode societal biases present in training data and generation can amplify or mask them depending on prompts and decoding. In cross-lingual tasks, gender and honorific systems introduce additional fairness dimensions [66]. While this thesis is methodological, the terminology of *bias*, *fairness*, and *harmful content* appears in evaluation contexts [67]. We use these terms in their standard NLP sense without entering normative debates.

Privacy and data provenance. Pretraining on web-scale corpora raises questions about personal data, scraping consent, and the right to be forgotten. Method descriptions that rely on external corpora should note provenance and access conditions where relevant [68]. Training-data extraction attacks show that large language models can memorize and regurgitate verbatim snippets, including personally identifiable information, underscoring the need for provenance and governance [69].

Safety. Alignment methods (instruction tuning, RLHF) are often motivated by safety and usability. In this section, we use *safety* to denote the avoidance of clearly harmful outputs under typical prompts [70]. Recent "constitutional" and related approaches demonstrate safety oriented fine-tuning without extensive human labeling, complementing RLHF [71]. Formal threat models are outside our scope.

1.1.14 Summary of key definitions

To close, we summarize the main definitions introduced here, deliberately without pointing forward to any specific methodology:

- **Token** x_i: an element of a discrete vocabulary where sequences $x_{1:L}$ feed the model.
- **Subword tokenization**: segmentation into morpheme-like units via BPE or unigram LM to manage OOVs and morphology.
- **Embedding** $e_i \in \mathbb{R}^d$: a learned vector representing token x_i as input to contextual layers.
- **Self-attention** (1.1): content-based weighting of positions, with multi-head variants operating in parallel.
- **Positional encodings**: mechanisms (absolute, relative, rotary) to inject order into attention-based models.
- **AR objective** (1.3): perplexity of next-token prediction for decoder-only LMs, supporting conditional generation.
- **MLM objective** (1.4): masked token prediction for encoder-only LMs.
- **Decoding**: procedures to map token distributions to sequences (greedy, beam, top-k, nucleus, constrained).

- **Overlap metrics** (1.5): ROUGE/BLEU-style scores based on n-gram statistics which is useful but imperfect.
- **PPL** (1.6): perplexity or the exponential of average negative log-likelihood per token.
- **PEFT**: adapters, BitFit, LoRA, and prefixes that adapt models with small, trainable modules.
- **MTL primitives**: task sampling policies, conditioning signals, and partial freezing to manage interference and sharing.
- **Ambiguity and preference elicitation**: recognizing that multiple correct outputs exist and clarifying latent user choices before decoding.
- **Long-sequence strategies**: sparse attention, kernelized attention, memory tokens, retrieval, SSMs, and hybrid designs that scale context.
- **Content selection**: building compact conditioning views (e.g., salient sentences) to focus generation on important material.
- **Reproducibility**: practices for reporting, code/data availability, and evaluation hygiene to support reliable comparisons.

The concepts above are intentionally decoupled from any one modeling pipeline. Their role is to provide a shared lexicon and minimal mathematical scaffolding that will make technical sections readable on their own terms, without presupposing familiarity with specific NLP subcultures or idiosyncratic notation. Subsequent sections will state concrete problem settings, objectives, and evaluation designs. The present section is limited to *what* the building blocks are and *how* they are conventionally defined.

1.2 Formal problem statement and scope

This section specifies the four problem settings addressed in the thesis and delimits their scope. For each, we formalize inputs/outputs, learning objectives, budgets/constraints, and the precise role of any intermediate signals. To keep this section purely problem-centric, datasets, training schedules, and empirical claims are deferred to later chapters. The evaluation protocols referenced are standard and summarized in Section 1.1.

1.2.1 Long-document abstractive summarization

Setting. Let $x = (x_1, \ldots, x_n)$ be a source document and $y = (y_1, \ldots, y_m)$ be a target abstractive summary with $m \ll n$. The model receives x and must produce y in free-form natural language.

Objective. A conditional generator with parameters θ minimizes sequence-level negative log-likelihood:

$$\hat{y} \;=\; \arg\max_{y} \; p_\theta(y \mid x), \qquad \mathcal{L}_{\mathrm{NLL}}(\theta) \;=\; -\sum_{(x,y)} \sum_{t=1}^{|y|} \log p_\theta\Big(y_t \,\Big|\, y_{<t}, x\Big). \tag{1.14}$$

Selective conditioning. To bound compute and focus content, a preselection map $s : \mathcal{X} \to \mathcal{X}$ returns a compact view $s(x)$ (e.g., top-k sentences or spans), subject to a budget $|s(x)| \leq K$. The generator is conditioned on $s(x)$:

$$\hat{y} \;=\; \arg\max_{y} \; p_\theta\Big(y \mid s(x)\Big), \qquad \mathcal{L}^{\mathrm{sel}}_{\mathrm{NLL}}(\theta) \;=\; -\sum_{(x,y)} \sum_{t=1}^{|y|} \log p_\theta\Big(y_t \,\Big|\, y_{<t}, s(x)\Big). \tag{1.15}$$

Scope constraints. (i) Documents may exceed a single forward-pass window – $s(x)$ is computed from x but the generator only sees $s(x)$ (and optionally coarse structure tags). (ii) No copy/pointer mechanism is assumed in the generator unless explicitly stated, allowing generation to be fully abstractive. (iii) The preselection budget K and sequence ordering used at training/inference are fixed a priori and reported with datasets.

Why this formulation. The problem is motivated by *abstractive fidelity vs. copying at long context* and by *metric–human misalignment.* Overlap metrics such as ROUGE can reward surface reuse. For example, purely extractive systems inflate scores without genuine abstraction, while naïvely abstractive systems may sacrifice coverage to avoid copying. By separating *content selection* from *realization*, constraining the conditioning signal to $s(x)$, and keeping the learning objective likelihood-based, the formulation encourages faithful abstraction without optimizing directly for overlap surrogates. This makes room for evaluation that jointly considers content coverage and human-judged qualities (fluency, coherence) without baking metric-specific biases into the objective.

Research questions. RQ$_1$: Does selective conditioning reduce n-gram copying (CopyRatio$_n$) while improving ROUGE-n on long document corpora? RQ$_2$: How do automatic metrics relate to human judgments (coherence/fluency vs. informativeness/relevance) under selective conditioning?

Chapter 2 instantiates $s(\cdot)$ with simple neural/heuristic extractors and a fixed input ordering that places conditioning content before the summary. The learning objective remains equation 1.15. Method details are deferred to that chapter.

1.2.2 Parameter-efficient multi-task learning

Setting. Let $\mathcal{T}$ be a finite set of supervised tasks. For each $t \in \mathcal{T}$, let $D_t = \{(x, y)\}$ be labeled data, and let f_θ denote a pretrained backbone (e.g., Transformer). The aim is to adapt to all $t \in \mathcal{T}$ under a *parameter budget* and *data-update budget*, while limiting interference across tasks.

Parameterization and budgets. Introduce small task-conditioned modules ϕ_t (e.g., adapters, low-rank updates, prefixes) and optionally unfreeze a subset of backbone weights. Let $m_\theta \in \{0, 1\}^{|\theta|}$ be a binary mask with $m_{\theta,i} = 1$ indicating frozen weights. The trainable parameter set is

$$\Theta_{\text{upd}} \;=\; \{\phi_t : t \in \mathcal{T}\} \;\cup\; \{\theta_i : m_{\theta,i} = 0\}, \qquad \|\Theta_{\text{upd}}\|_0 \le B_{\text{param}},$$

where B_{param} is a hard budget on the number of trainable parameters (or an equivalent memory budget). A data-update budget may cap gradient-bearing examples per task.

Objective and sampling. With a task-sampling distribution q over $\mathcal{T}$,

$$\min_{\theta,\{\phi_t\}} \;\; \mathbb{E}_{t\sim q}\; \mathbb{E}_{(x,y)\sim D_t}\; \ell\Big(f_{\theta,\phi_t}(x), y\Big) \quad \text{s.t.} \quad \|\Theta_{\text{upd}}\|_0 \le B_{\text{param}}. \tag{1.16}$$

Here ℓ is the task-appropriate supervised loss (e.g., cross-entropy) and q is fixed or parameterized (e.g., temperature-scaled by dataset size).

Scope constraints. (i) No task-specific encoders are introduced. All tasks share f_θ plus the small ϕ_t. (ii) Task conditioning is provided via explicit IDs or learned embeddings. No task-specific vocabularies are assumed beyond output heads. (iii) The choice of PEFT mechanism is orthogonal to equation 1.16. The problem statement only requires adherence to budgets.

Why this formulation? The problem targets *parameter inefficiency & forgetting in MTL* and *data imbalance across tasks*. Full fine-tuning per task multiplies storage/serving footprints and increases the risk of catastrophic forgetting when tasks are later mixed. Constraining $\|\Theta_{\text{upd}}\|_0$ forces solutions that adapt with few trainables and limits drift by freezing most of f_θ. Realistic multi-task corpora are skewed. Exposing $q(t)$ as a first-class variable acknowledges that uniform or size-proportional sampling can under-serve low-resource tasks or overfit high-resource ones, and that principled schedules are part of the problem's definition rather than an afterthought.

Research question. RQ$_3$: Can task-conditioned adapters with partial freezing and principled sampling exceed strong baselines on multi-task suites (e.g., GLUE) with fewer trainable parameters and fewer updates?

Chapter 3 chooses concrete ϕ_t (conditional adapters, conditional attention biases, condition layer norm) and a sampling policy q, which evaluates equation 1.16 under fixed budgets without altering the objective.

1.2.3 Crosslingual generation with ambiguity resolution

Setting. Let $x^{(s)}$ be a source-language sentence and $y^{(t)}$ a target-language rendering. Due to underspecification, multiple $y^{(t)}$ may be acceptable depending on latent user preferences U (e.g., formality, referent properties, lexical sense). The model is permitted to query a user surrogate before generating.

Interaction budget. Allow J pre-generation interactions $\tau = \{(q_j, a_j)\}_{j=1}^{J}$, where $q_j \in \mathcal{Q}$ are clarifying questions and a_j are answers produced by a user model conditioned on $(x^{(s)}, U, q_{1:j})$. The budget J is fixed by protocol.

Objective. A conditional generator p_ψ produces y given $(x^{(s)}, \tau)$:

$$\hat{y} \;=\; \arg\max_{y}\, p_\psi\Big(y \mid x^{(s)}, \tau\Big). \tag{1.17}$$

The value of acquiring τ can be formalized via (a) *downstream risk* minimization after J interactions,

$$\min_{\pi\in\Pi}\; \mathbb{E}_{U,x^{(s)}}\; \mathbb{E}_{\tau\sim\pi(\cdot|x^{(s)})}\; \mathbb{E}_{y\sim p_\psi(\cdot|x^{(s)},\tau)}\Big[\,\ell(y; U, x^{(s)})\,\Big], \tag{1.18}$$

where π is a question policy and ℓ a task-appropriate evaluation loss or (b) an *information gain* surrogate that encourages resolving U:

$$\max_{\pi\in\Pi}\; \mathbb{E}_{x^{(s)}}\; \mathbb{E}_{\tau\sim\pi(\cdot|x^{(s)})}\Big[\, I\Big(U; \tau \mid x^{(s)}\Big)\,\Big] \;=\; \max_{\pi}\; \mathbb{E}_{x^{(s)},\tau}\, \mathrm{KL}\Big(p(u \mid x^{(s)}, \tau) \,\|\, p(u \mid x^{(s)})\Big). \tag{1.19}$$

Either view defines the problem without committing to a particular acquisition rule or answer model.

Scope constraints. (i) All clarifications occur *before* decoding. The generator does not query during generation. (ii) The user model is protocol-dependent and treated as exogenous to p_ψ. (iii) Primary evaluation is translation quality under fixed J and fixed ambiguity types. Metric choice follows Section 1.1.

Why this formulation? The problem directly addresses *ambiguity without elicitation*. Many translation errors arise not from modeling deficits but from hidden preferences (register, referent attributes) that the source text does not specify. Allocating a pre-decoding interaction budget and formalizing either risk reduction equation 1.18 or information gain equation 1.19 makes preference surfacing a first-class objective rather than an incidental prompt-engineering trick. This also mitigates *metric–human misalignment*: by conditioning on elicited preferences, the target distribution better matches what human raters deem appropriate even when overlap-based metrics are insensitive to such choices.

Research question. RQ$_4$: How does a pre-decoding elicitation chain affect $I(U;\tau \mid x^{(s)})$ and downstream translation choices?

Chapter 4 instantiates π as a prompt-driven policy, fixes J, and operationalizes equation 1.17–equation 1.19 on targeted ambiguity phenomena and language pairs.

1.2.4 Hybrid long-range language modeling

Setting. Consider next-token language modeling over long sequences $x_{1:L}$ with L large enough that quadratic attention becomes a bottleneck. The goal is to specify a layer class that (i) preserves content-sensitive local modeling and (ii) provides efficient long-range contextualization with subquadratic cost, while remaining *fully parallelizable* across tokens within a training step.

Layer composition. Let $\mathcal{F}_{\text{SSM}}$ denote a 1-D convolutional sublayer induced by a (possibly structured) state-space model and $\mathcal{F}_{\text{BlkAttn}}$ a block-wise attention sublayer over windows of size W. A hybrid layer computes

$$\mathbf{Y}^{\text{SSM}} \;=\; \mathcal{F}_{\text{SSM}}(\mathbf{X}), \qquad \mathbf{Y}^{\text{Attn}} \;=\; \mathcal{F}_{\text{BlkAttn}}(\mathbf{X}; W), \qquad \mathbf{Y} \;=\; \Gamma\Big([\mathbf{Y}^{\text{SSM}}, \mathbf{Y}^{\text{Attn}}]\Big), \tag{1.20}$$

where Γ is a fusion map (e.g., concatenation + projection or gated sum). The SSM branch provides long-range, content-agnostic propagation and the block-attention branch provides local, content-aware mixing.

Complexity target and constraints. The design target is *additive* cost

$$\mathcal{C}(L, W) \;=\; O(L \log L) \;+\; O(W^2),$$

stemming from FFT-based SSM convolution and windowed attention, respectively. Additional constraints are: (i) causality in both branches (no access to future tokens), (ii) parallel token processing (no explicit recurrent loops across windows in the hybrid layer), and (iii) compatibility with standard optimizer states and training schedules for AR LMs.

Language modeling objective. The hybrid layer class is used inside a decoder-only architecture that minimizes

$$\min_{\Theta} \ \mathbb{E}_{x_{1:L}}\Big[-\sum_{t=1}^{L} \log p_{\Theta}(x_t \mid x_{<t})\Big], \tag{1.21}$$

with the only difference from conventional stacks being the replacement of some attention blocks by equation 1.20 or interleaving them.

Why this formulation? The problem addresses the *SSM Transformer gap under runtime constraints*. Attention affords precise, content-addressable interactions but scales quadratically in L. SSMs propagate information in (near) linear time but lack fine-grained content lookup. By composing these biases within a single layer and enforcing additive complexity and full parallelism, the formulation defines a design envelope appropriate for very long contexts without sacrificing local precision or practical trainability.

Research question. RQ$_5$: Can an SSM+attention hybrid deliver improved perplexity and practical speed at long lengths while maintaining full parallelizability?

Chapter 5 instantiates equation 1.20 with specific SSM parameterizations and concrete fusion rules, fixes W, and reports performance/efficiency under equation 1.21.

1.3 Thesis structure and guiding threads

The remainder of this dissertation follows the following line of thought: *structure the problem so that model capacity is spent where it matters.* The chapters move from structuring the *input* (what to show the model), to structuring the *parameters* (what to adapt and what to freeze), to structuring the *interaction* (what to ask before generating), and finally to structuring the *layer* (how to carry long-range context efficiently). The intent is not to multiply formalisms already stated, but to explain how each article occupies one step in this program and how the steps speak to one another as a coherent whole[1].

[1] Please note that I have used Codex to integrate various articles with specific conference's Latex templates into the single Polytechnique Montréal Latex template. Additionally, I have used Grok and GPT-5 for reviewing, feedback, thesis planning, fixing typos, and researching papers.

Chapter 1 Orientation and shared background. Chapter 1 (this chapter) motivates the learning-based stance, sets notation, and gathers the definitions needed later (Transformers, pretraining objectives, decoding, evaluation signals, parameter-efficient fine-tuning, long-sequence strategies). It also frames the practical issues that recur across the thesis long inputs, multi-task deployment, underspecified user intent, and long-range modeling under runtime constraints so that subsequent chapters can focus on solutions rather than preliminaries.

Chapter 2: Selective conditioning for long-document summarization. We begin with long-document summarization, where the question is how to encourage genuine *abstraction* rather than surface reuse when inputs exceed typical context windows. Chapter 2 introduces a small, budgeted pre-selection step placed in front of a Transformer generator so that the model need not "fight the entire document at once, " and the summary can be conditioned on just what matters. The core contribution is to show that this selective conditioning reduces n-gram copying while improving overlap metrics and to read those scores alongside human judgments (coherence, fluency, informativeness, relevance). In the language of research questions, this chapter addresses: RQ 1.2.1 (copying vs. ROUGE-n) and RQ 1.2.1 (metric–human relationships). Methodologically, the move is modest extract, then condition but it sets up a pattern that recurs throughout the thesis: clarify the target and constraints before asking the model to generate.

Chapter 3: Parameter-/data-efficient multi-task transfer. Next we turn from structuring inputs to structuring parameters. When a single backbone must serve many tasks, fully fine-tuning per task becomes costly and brittle, especially when joint training can invite interference. Chapter 3 keeps most of the pretrained model fixed and routes task-specific variation through small, task-conditioned modules, while a principled sampling policy controls which tasks see gradients when. The contribution is a demonstration that such parameter-/data-efficient sharing can exceed strong baselines on multi-task suites with fewer trainable parameters and fewer updates, addressing RQ 1.2.2. Intuitively, the chapter separates *what is shared* from *what is adapted*, much as Chapter 2 separated *what is selected* from *what is generated.*

Chapter 4: Interactive chains for ambiguity-aware translation. The third step shifts the focus from system internals to protocol: before a model produces text in settings like machine translation, users often have latent preferences (register, referent properties, sense choices) that are not spelled out in the source. Rather than relying on longer prompts alone, Chapter 4 allocates a short, pre-decoding interaction to elicit those preferences through targeted questions, then conditions generation on the answers. The contribution is a procedural account of ambiguity resolution an *interactive chain* and an evaluation that shows how this reduces uncertainty about user intent and improves downstream decisions, addressing RQ 1.2.3. Conceptually, this mirrors the earlier steps: once again, we reduce uncertainty *before* decoding, but here by engaging the user rather than filtering the input or re-wiring parameters.

Chapter 5: Block-State Transformers for long-range modeling. Finally, we address the cost of carrying context across long sequences. Attention offers precise, content-addressable mixing but becomes expensive as sequences grow while state-space models (SSMs) propagate information efficiently but lack fine-grained lookup. Chapter 5 composes the two at the layer level, running an SSM pathway in parallel with a block-attention pathway and fusing the results under an additive complexity target. The contribution is an empirical case that such a hybrid can improve perplexity and wall-clock behavior at long lengths while retaining full token-parallel training, addressing RQ 1.2.4. We also show strong performance on Long Range Arena tasks under comparable parameter budgets. In the broader program, this is the architectural counterpart to the earlier structuring ideas: after structuring inputs, parameters, and protocol, we now structure the layer itself to better match the problem's computational shape.

Chapter 6: Synthesis and outlook. The closing chapter does not propose a single master algorithm. It traces a design pattern that has emerged across the articles: *pre-select, condition, interact, hybridize.* Pre-select what the model sees (Chapter 2). Condition the backbone with small, task-aware pathways (Chapter 3). Interact briefly to elicit missing preferences (Chapter 4). Hybridize mechanisms to reconcile long-range efficiency with local precision (Chapter 5)). The synthesis also reflects on limitations observed along the way where metric–human alignment is fragile, how sampling and interaction policies might be made more principled, and what it would take to extend these ideas to retrieval-augmented or tool-use scenarios.

CHAPTER 2 ARTICLE 1: ON EXTRACTIVE AND ABSTRACTIVE NEURAL DOCUMENT SUMMARIZATION WITH TRANSFORMER LANGUAGE MODELS

Jonathan Pilault[1,2,3,*], Raymond Li[1,*], Sandeep Subramanian[1,2,4,*],
and Christopher Pal[1,2,3,4,5]

Accepted in 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Publication date: 11/18/2020

[1]Element AI [2]Mila [3]Polytechnique Montreal
[4]University of Montreal [5]Canada CIFAR AI Chair

[*] Authors contributed equally to this work[1].

Abstract

We present a method to produce abstractive summaries of long documents that exceed several thousand words via neural abstractive summarization. We perform a simple extractive step before generating a summary, which is then used to condition the transformer language model on relevant information before being tasked with generating a summary. We also show that this approach produces more abstractive summaries compared to prior work that employs a copy mechanism while still achieving higher ROUGE scores. We provide extensive comparisons with strong baseline methods, prior state of the art work as well as multiple variants of our approach including those using only transformers, only extractive techniques and combinations of the two. We examine these models using four different summarization tasks and datasets: arXiv papers, PubMed papers, the Newsroom and BigPatent datasets. We find that transformer based methods produce summaries with fewer n-gram copies, leading to n-gram copying statistics that are more similar to human generated abstracts. We include a human evaluation, finding that transformers are ranked highly for coherence and fluency, but purely extractive methods score higher for informativeness and relevance. We hope that these architectures and experiments may serve as strong points of comparison for future work[2].

[1] Jonathan Pilault a contribué dans la conception des algorithmes, l'experimentation, l'analyse des résultats, amélioration de l'efficacité du logiciel, analyse théorique, rédaction de l'article, recherche bibliographique et la réponse aux réviseurs de conférence.

[2] Note: The abstract above was collaboratively written by the authors and one of the models presented in this paper based on an earlier draft of this paper.

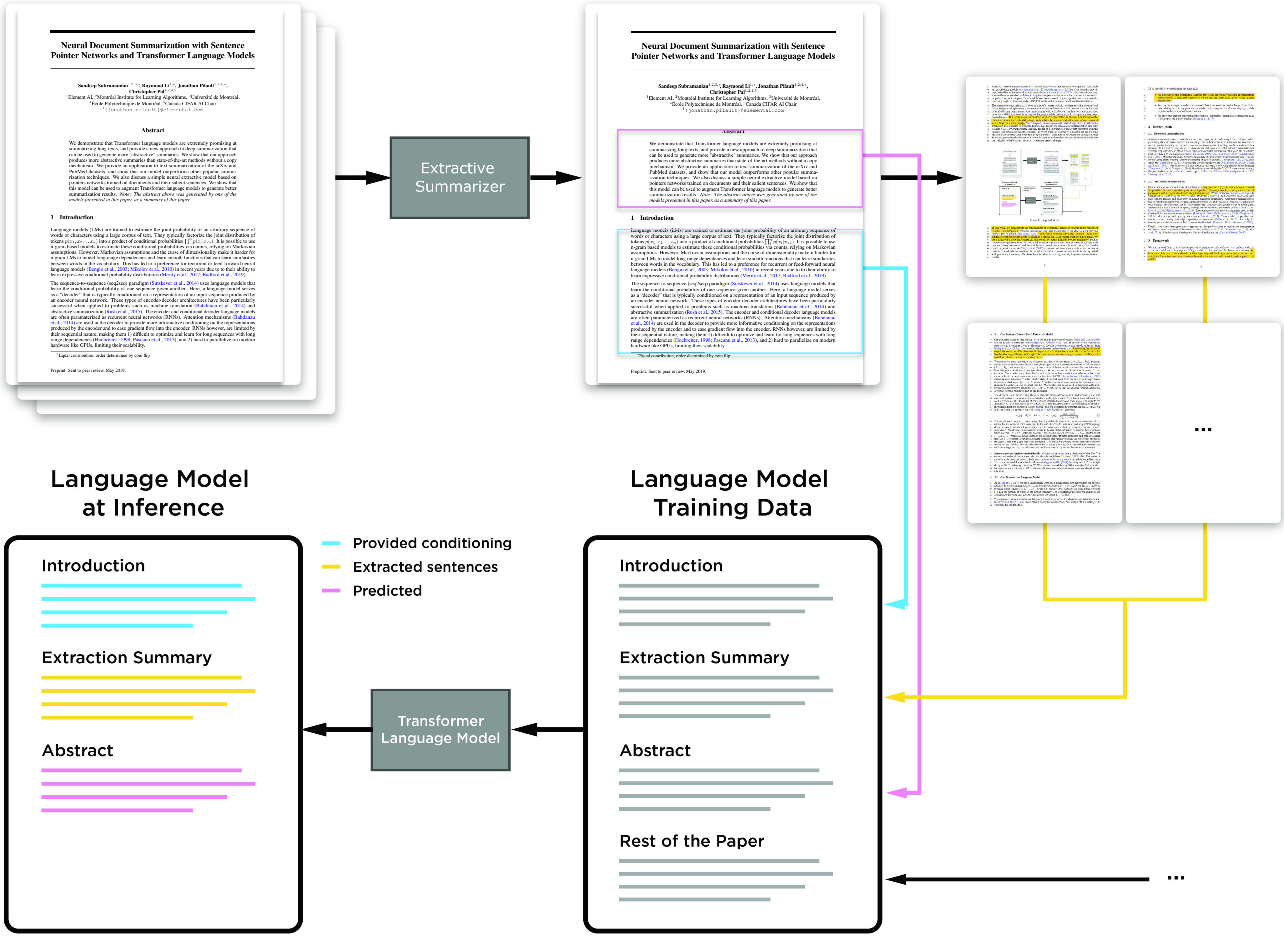


Figure 2.1 Our approach for abstractive summarization of a scientific article. An older version of this paper is shown as the reference document. First, a sentence pointer network extracts important sentences from the paper. Next, these sentences are provided along with the whole scientific article to be arranged in the following order: Introduction, extracted Sentences, abstract & the rest of the paper. A transformer language model is trained on articles organized in this format. During inference, the introduction and the extracted sentences are given to the language model as context to generate a summary. In domains like news and patent documents, the introduction can be replaced by the entire document.

2.1 Introduction

Automatic text summarization is the process of compressing a document while preserving key information content and meaning. This process is often achieved through extractive or abstractive techniques. Extractive summarization is the strategy of selecting a subset of words, phrases or sentences from the input document to form a summary. Abstractive summarization consists of creating sentences summarizing content and capturing key ideas and elements of the source text, usually involving significant changes and paraphrases of text from the original source sentences. While extractive summarization is able to preserve saliency, the broader flow or coherency of the multiple sentences forming the summary can be less natural compared to a human generated summary. On the other hand, abstractive methods should produce coherent summaries without copying sentences verbatim while remaining faithful to statements asserted in the input document.

Recent work by [72] (GPT-2) has demonstrated that Transformer Language Models (TLMs) trained on web text can inadvertently learn to perform abstractive summarization, since a large crawl of web documents may contain some documents which have a “tl;dr” token followed by a summary. We are interested here in explicitly configuring autoregressive transformer models to generate summaries in an intentional and focused manner. Since summaries or abstracts typically appear at the beginning of a document, a model trained from such web-crawl data does not enforce strong conditioning on the text to be summarized. Our tests using models naively trained on web-crawl data yielded summarization quality far below baseline methods. However, in this paper we explore what can be achieved through simply ordering the passages of an input text, correctly structuring the task definition and training procedure. We also examine the impact of combining this approach with simple but high quality extractive techniques.

While pure language models can be applied to short input documents, memory considerations make it difficult to scale to long documents. Further, as high quality extractive summarization methods illustrate, much of the content of a long document is not needed to create a summary. For these reasons we also explore a hybrid approach which combines an extractive and abstractive approach. We achieve this by stepping away from the classical end-to-end sequence-to-sequence paradigm, using an initial extractive step that reduces the amount of context for a subsequent abstractive step (see figure 2.1). Such an approach could be thought of as a form of hard attention. Moreover, we show that such a paradigm works even for datasets where the entire input can fit in memory, i.e. see Table 2.4 and 2.5. We take an approach whereby we restructure the input to a TLM by reordering the document and inserting standardized delimiters to identify the introduction, our extracted sentences, the abstract or summary and the rest-of-the-article. With our method, the resulting TLM can focus its attention on the relevant content and its model complexity on the summarization task.

In general, as we shall detail in our experiments below, we find that TLMs are surprisingly effective at summarizing *long documents*, outperforming typical seq2seq approaches, even without using copying/pointing mechanisms, an encoder or additional losses. Our contribution consists of an extensive set of large scale experiments comparing our hybrid extractive and abstractive approach to long document summarization with different variants of our model, strong and simple baselines as well as with state-of-the-art summarization models (see section 2.3.2 for a complete description of comparisons). We examine these models through ROUGE scores, through a study of the amount of n-gram copying performed by different models, as well as through a human evaluation using a standard protocol. We find that our hybrid approach yields results that surpass current state-of-the-art results on several metrics of these evaluations.

We see our extensive experimentation and the wide variety of evaluation protocols provided here as being a key part of the contribution provided by this work and we hope that the analysis, insights and models here will serve as strong yet simple baselines for future comparison and research.

2.2 Related Work

The earliest attempts at automatic summarization focused on extractive techniques, which find words or sentences in a document that capture its most salient content. Recently, with advances in distributed representations of words, phrases and sentences, researchers have proposed to use these to compute similarity scores. Such techniques were further refined by [73, 74, 75] with encoder-decoder architectures - the representations learned by the encoder are used to choose the most salient sentences. [74] and [73] trained encoder-decoder neural networks as a binary classifier to determine if each sentence in a document should belong to the extractive summary or not. [75] use a pointer network [76] to sequentially pick sentences from the document that comprise its extractive summary. Such techniques however heavily rely on the span of words from the input document.

Human summarizers have four common characteristics. They are able to (1) interpret a source document, (2) prioritize the most important parts of the input text, (3) paraphrase key concepts into coherent paragraphs and (4) generate diverse output summaries. While extractive methods are arguably well suited for identifying the most relevant information, such techniques may lack the fluency and coherency of human generated summaries. Abstractive summarization has shown the most promise towards addressing points (3) and (4) above. Abstractive generation may produce sentences not seen in the original input document. Motivated by neural network success in machine translation experiments, the attention-based encoder decoder paradigm has recently been widely studied in abstractive summarization [77, 78, 79]. The advantages of extractive, abstractive and attention-based models were first combined in [80, 81] with a copy mechanism for out-of-vocabulary words present in the source document. Similarly, [82] used the attention scores to calculate the probability of generating vs copying a word.

The most similar approach to our hybrid extractive and abstractive technique is that of [75, 83, 84, 85]. In such set-ups, an *extractor* first selects salient sentences from the input. Then, an abstractive summarizer rewrites extracted sentences into a final summary. Our framework has a few advantages over previous methods. 1), we explore high capacity transformer LMs akin to [72] as our abstractive summarizer, which results in grammatical and fluent generations 2), our language modeling formulation of the problem allows us to easily "recycle" the input document and use it as additional in-domain data for LM training. 3) We improve over previous approaches without the use of a copy mechanism, which results in fewer n-gram copies from the input document. [85] generate Wikipedia articles given references to source material and extracted sentences. They rank the importance of paragraphs found in the reference material based on techniques such as TextRank [86], a graph based ranking technique. In contrast, the extractive methods we use here are trained discriminatively using an extractive abstract as the target that is generated using an oracle. Wikipedia article synthesis also necessarily combines potentially redundant information from multiple documents that is relatively specific and less abstractive compared to the task of writing the abstract of a scientific paper. As seen in Figure 2.2, human generated (ground-truth) abstractive summaries in our datasets actually have very little word overlap with the source document.

2.3 Framework

Our model comprises two distinct trainable components: 1) an extractive model, comprising a hierarchical encoder that outputs sentence representations, used to either point to or classify sentences in the input, and 2) a transformer language model, conditioned on the extracted sentences as well as a part of or the entire input document.

2.3.1 Extractive Models

We describe the two neural extractive models used in this section. We used different types of extraction techniques to demonstrate the TLM model sensitivity to the extracted sentences. For instance, the Sentence Pointer performs much better on the arxiv dataset (see table 2.2) but the classifier is stronger on the Pubmed dataset (see table 2.3).

Hierarchical Seq2seq Sentence Pointer Our extractive model is similar to the sentence pointer architecture developed by [75] with the main difference being the choice of encoder. We use a hierarchical bidirectional LSTM encoder with word and sentence level LSTMs while [75] use a convolutional word level encoder for faster training and inference. The decoder is in both cases is an LSTM.

The procedure to determine ground-truth extraction targets is similar to previous work [87]: the ground truth is determined by computing the average $\text{ROUGE}_{1,2,L}$ score of each document sentence against each summary sentence. Considering the input document as a list of N sentences $D = (S_1, \ldots, S_N)$ and the target summary as a list of M sentences $T = (S'_1, \ldots, S'_M)$, our heuristic provides $N \times M$ scores, such that: $\text{SCORES}_{\text{extraction}} = \{\frac{1}{3}\sum_{r\in 1,2,L} \text{ROUGE}_r(S_i, S'_j) | S_i \in D; S'_j \in T\}$.

Since single sentence extraction may not always contain the same information content as a target summary, we extended the number ground-truth extraction sentences per output summary sentence to two. This is done by choosing the top 2 sentences in D that have the highest $\text{SCORES}_{\text{extraction}}$ with respect to a given sentence in T. The resulting $2M$ ordered sentences are used as context in the TLM. The TLM benefits from a more structured and larger context from the extractive summarization model during training.

First, the “sentence-encoder” or token-level RNN is a bi-directional LSTM [88] encoding each sentence. The last hidden state of the last layer from the two directions produces sentence embeddings: $(\mathbf{s}_1, \ldots, \mathbf{s}_N)$, where N is the number of sentences in the document. The sentence-level LSTM or the “document encoder”, another bi-directional LSTM, encodes this sequence of sentence embeddings to produce document representations: $(\mathbf{d}_1, \ldots, \mathbf{d}_N)$.

The decoder is an autoregressive pointer LSTM taking the sentence-level LSTM hidden state of the previously extracted sentence as input and predicting the next extracted sentence. Let i_t the index of the previous extracted sentence at time step t. The input to the decoder is $\mathbf{s}_{i_t}$.The decoder’s output is computed by an attention mechanism from the decoder’s hidden state $\mathbf{h}_t$ over the document representations $(\mathbf{d}_1, \ldots, \mathbf{d}_N)$. We used the dot product attention method from [89]. The attention weights $\mathbf{a}_t$ produce a context vector $\mathbf{c}_t$, which is then used to compute an attention aware hidden state $\tilde{\mathbf{h}}_t$.

The attention weights $\mathbf{a}_t$ are used as output probability distribution over the document sentences, of the choice for the next extracted sentence. The model is trained to minimize the cross-entropy of picking the correct sentence at each decoder time step. At inference, we use beam-search to generate the extracted summary.

Sentence Classifier As with the pointer network, we use a hierarchical LSTM to encode the document and produce a sequence of sentence representations $\mathbf{d}_1, ..., \mathbf{d}_N$ where N is the number of sentences in the document. We compute a final document representation as follows:

$$\mathbf{d} = \tanh\left(\mathbf{b}_d + \mathbf{W}_d \frac{1}{N}\sum_{i=1}^{N}\mathbf{d}_i\right) \tag{2.1}$$

where $\mathbf{b}_d$ and $\mathbf{W}_d$ are learnable parameters. Finally, the probability of each sentence belonging to the extractive summary is given by:

$$o_i = \sigma\left(\mathbf{W}_o \begin{bmatrix}\mathbf{d}_i \\ \mathbf{d}\end{bmatrix} + \mathbf{b}_o\right) \tag{2.2}$$

where σ is the sigmoid activation function. The model is trained to minimize the binary cross-entropy loss with respect to the sentences in the gold-extracted summary.

Model details and training parameters are included in the appendix.

2.3.2 Transformer Language Models (TLM)

Instead of formulating abstractive summarization as a seq2seq problem using an encoder-decoder architecture, we only use a single transformer language model that is trained *from scratch*, with appropriately "formatted" data (see figure 2.1, we also describe the formatting later in this section).

We use a transformer [90] language model (TLM) architecture identical to [72]. Our model has 220M parameters with 20 layers, 768 dimensional embeddings, 3072 dimensional position-wise MLPs and 12 attention heads. The only difference in our architectures (to our knowledge) is that we do not scale weights at initialization. We trained the language model for 5 days on 16 V100 GPUs on a single Nvidia DGX-2 box. We used a linear ramp-up learning rate schedule for the first $40,000$ updates, to maximum learning rate of $2.5 \times e^{-4}$ followed by a cosine annealing schedule to 0 over the next $200,000$ steps with the Adam optimizer. We used mixed-precision training [91] with a batch size of 256 sequences of 1024 tokens each.

In order to get an unconditional language model to do abstractive summarization, we can use the fact that LMs are trained by factorizing the joint distribution over words autoregressively. In other words, they typically factorize the joint distribution of tokens $p(x_1, x_2 \dots x_n)$ into a product of conditional probabilities $\prod_i^n p(x_i|x_{<i})$. We therefore organize the training data for our models such that the ground-truth summary *follows* the information used by the model to generate a summary. As such, we can model the joint distribution of the document and the summary during training, and sample from the conditional distribution of the summary given document when we wish to perform inference.

When dealing with extremely long documents that may not fit into a single window of tokens seen by a transformer language model, such as an entire scientific article, we use its introduction as a proxy for having enough information to generate an abstract (summary) and use the remainder of the paper as in domain language model training data (Fig 2.1). In such cases, we organize the arXiv and PubMed datasets as follows: 1) the paper introduction, 2) extracted sentences from the sentence pointer model, 3) the abstract, and 4) the rest of the paper. This ensures that at inference time, we can provide the language model the paper introduction and the extracted sentences as conditioning to generate its abstract. We found that using the ground truth extracted sentences during training and the model extracted sentences at inference performed better than using the model extracted sentences everywhere. On other datasets, the paper introduction would be the entire document. In such case, the rest of the paper does not exist and is therefore not included.

We use a special token to indicate the start of the summary and use it at test time to signal to the model to start generating the summary. The rest of the article is provided as additional in-domain training data for the LM. The entire dataset is segmented into non-overlapping examples of $1,024$ tokens each. We use "topk" sampling at inference [92, 72], with $k = 30$ and a softmax temperature of 0.7 to generate summaries.

2.4 Results and Analysis

Datasets We experiment with four different large-scale and long document summarization datasets - arXiv, PubMed [2], bigPatent [1] and Newsroom [4]. Statistics are reported in Table 2.1.

Data preprocessing Both our extractive and abstractive models use sub-word units computed using *byte pair encoding* [93] with $40,000$ replacements. To address memory issues in the sentence pointer network, we only keep 300 sentences per article, and 35 tokens per

Dataset	#Documents	Comp Ratio	Sum Len	Doc Len
arXiv	215,913	39.8	292.8	6,913.8
PubMed	133,215	16.2	214.4	3,224.4
Newsroom	1,212,726	43.0	30.4	750.9
BigPatent	1,341,362	36.4	116.5	3,572.8

Table 2.1 Statistics from [1] for the datasets used in this work - The number of document/-summary pairs, the ratio of the number of words in the document to the abstract and the number of words in the summary and document.

sentence.

Evaluation We evaluate our method using full-length F-1 ROUGE scores [94] and re-used the code from [2] for this purpose. All ROUGE numbers reported in this work have a 95% confidence interval of at most 0.24.

Comparison We compare our results to several previously proposed extractive, abstractive and mixed summarization models on ROUGE scores. ROUGE scores tend to measure lexical overlap [95] which favors extractive methods of summarization. Since ROUGE scores do not capture system summary fluency and readability (which typically does not favor abstractive summarization), we also include a human evaluation. For this reason, Tables 2.2, 2.3, 2.4, 2.5 have a “Type” column to inform the reader on the type model evaluated (Ext=extractive, Mix=mixed and Abs=abstractive). All prior results reported on the arXiv and Pubmed benchmark are obtained from [2], except for the *Bottom-up* model[3] [83]. Similarly, prior results for the BigPatent dataset are obtained from [1] and Newsroom from [4] and [5]. These methods include *LexRank* [96], *SumBasic* [97], *LSA* [98], *Attention-Seq2Seq* [78, 79], *Pointer-Generator Seq2Seq* [82], *Discourse-aware*, which is a hierarchical extension to the pointer generator model, [2], *Sent-rewriting* [75], *RNN-Ext* [75], *Exconsumm* [5].

We present our main results on summarizing arXiv and PubMed papers in tables 2.2, 2.3. TLM+I+E (G,M) sets a new state-of-the-art on Arxiv, Pubmed and bigPatent datasets on abstractive summarization ROUGE scores. Our extractive models are able to outperform previous extractive baselines on both the arXiv and Pubmed datasets. Our extractive techniques also score higher than our abstractive techniques on arXiv and Pubmed. Again, ROUGE does not capture all aspects of a summary’s quality such as fluency and coherence. For instance, previous work that have used RL to maximize ROUGE scores have concluded that "RL has the highest ROUGE-1 and ROUGE-L scores, it produces the least readable

[3] We used the code from `https://github.com/sebastianGehrmann/bottom-up-summary` with the same parameters.

summaries" [99]. Our TLM conditioned on the extractive summary produced by our best extractive model (TLM-I+E (G,M)) outperforms prior abstractive/mixed results on the arXiv, Pubmed and bigPatent datasets, except on ROUGE-L.

Model	Type	ROUGE			
		1	2	3	L
Previous Work					
Lead-10	Ext	35.52	10.33	3.74	31.44
SumBasic	Ext	29.47	6.95	2.36	26.3
LexRank	Ext	33.85	10.73	4.54	28.99
Seq2Seq	Abs	29.3	6.00	1.77	25.56
Pointer-gen	Mix	32.06	9.04	2.15	25.16
Discourse-aware	Mix	35.80	11.05	3.62	31.80
Bottom-up	Mix	39.96	13.16	5.04	36.28
Our Models					
Sent-CLF	Ext	34.01	8.71	2.99	30.41
Sent-PTR	Ext	42.32	15.63	7.49	38.06
TLM-I	Abs	39.65	12.15	4.40	35.76
TLM-I+E (G,M)	Mix	41.62	14.69	6.16	38.03
Oracle					
Gold Ext	Oracle	44.25	18.17	9.14	35.33
TLM-I+E (G,G)	Oracle	46.40	18.15	8.71	42.27

Table 2.2 Summarization results on the arXiv dataset. Previous work results from [2]. The following lines are a simple baseline Lead-10 extractor and the pointer and classifier models. Our transformer LMs (TLM) are conditioned either on the Introduction (I) or along with extracted sentences (E) either from ground-truth (G) or model (M) extracts.

On Newsroom, our TLM model performs close to 7 times better than the other purely abstractive model (Seq2Seq with attention). We achieve better performance than the pointer generator even on the abstractive and mixed which their model should be better suited for since it has a copy mechanism. The Exconsumm model [5] however, which is primarily an extractive model does better on this dataset. We suspect the poor ROUGE-L result is due to the absence of a copy mechanism that makes it hard to get *exact* large n-gram matches. Figure 2.2 further supports this hypothesis, it is evident that a model with a copy mechanism is often able to copy even upto 25-grams from the article. Further, [100] finds that ROUGE-L is poorly correlated with human judgements when compared to ROUGE-1,2,3. In tables 2.8, 2.9 A.1, we present qualitative results of abstracts of notable papers in our field and of our TLM conditioned on the introductions and extracted summaries of a ran-

Model	Type	ROUGE			
		1	2	3	L
Previous Work					
Lead-10	Ext	37.45	14.19	8.26	34.07
SumBasic	Ext	37.15	11.36	5.42	33.43
LexRank	Ext	39.19	13.89	7.27	34.59
Seq2seq	Abs	31.55	8.52	7.05	27.38
Pointer-gen	Mix	35.86	10.22	7.60	29.69
Discourse-aware	Mix	38.93	15.37	9.97	35.21
Bottom-up	Mix	40.02	15.82	8.71	37.28
Our Models					
Sent-CLF	Ext	45.01	19.91	12.13	41.16
Sent-PTR	Ext	43.30	17.92	10.67	39.47
TLM-I	Abs	37.06	11.69	5.31	34.27
TLM-I+E (G,M)	Mix	42.13	16.27	8.82	39.21
Oracle					
Gold Ext	Oracle	47.76	20.36	11.52	39.19
TLM-I+E (G,G)	Oracle	46.32	20.15	11.75	43.23

Table 2.3 Summarization results on the PubMed dataset. Previous work results from [2]. The following lines are a simple baseline Lead-10 extractor and the pointer and classifier models. Our transformer LMs (TLM) are conditioned either on the Introduction (I) or along with extracted sentences (E) either from ground-truth (G) or model (M) extracts.

dom example from the arXiv test set. Table 2.6 shows similar qualitative examples on the Newsroom dataset. Tables 2.2, 2.3 and 2.4 also provide different train / test settings for our TLM conditioned on extracted sentences. We show a performance upper bound conditioning the Transformer LM on oracle / ground-truth extracted sentences at both train and test time (TLM-I+E (G,G)). We also experiment with using either the ground-truth extracted sentences (TLM-I+E (G,M)) or the model extracted sentences (TLM-I+E (M,M)) during training and find that latter slightly impairs performance. It is important to note that, across datasets, introducing extracted sentences with TLM+I+E or TLM+E has consistently performed better over TLM+I or TLM. For bigPatent in table 2.4 and newsroom in table 2.5 TLM and TLM+E models have access to the same text since the whole article can fit in the transformer window size. This is particularly interesting since our results show that explicitly delimiting the extracted sentences has large positive affects on summary performance. As anticipated, introducing extracted sentences allows the TLM model to focus less on information retrieval and more on language generation.

Model	Type	ROUGE		
		1	2	L
Previous Work				
Lead-3	Ext	31.27	8.75	26.18
TextRank	Ext	35.99	11.14	29.60
LexRank	Ext	35.57	10.47	29.03
RNN-Ext	Ext	34.63	10.62	29.43
Seq2Seq	Abs	28.74	7.87	24.66
Pointer-gen	Mix	30.59	10.01	25.65
Pointer-gen (Cov)	Mix	33.14	11.63	28.55
Sent-rewriting	Mix	37.12	11.87	32.45
Our Models				
Sent-CLF	Ext	36.20	10.99	31.83
Sent-PTR	Ext	34.21	10.78	30.07
TLM	Abs	36.41	11.38	30.88
TLM+E (G,M)	Mix	**38.65**	**12.31**	**34.09**
Oracle				
Gold Ext	Oracle	43.56	16.91	36.52
OracleFrag	Oracle	91.85	78.66	91.85
TLM+E (G,G)	Oracle	39.99	13.79	35.33

Table 2.4 Summarization results on the bigPatent dataset. Previous work results from [1]. Our transformer LMs (TLM) are conditioned on the whole document or additionally with extracted sentences (E) either from ground-truth (G) or model (M) extracts. Note that OracleFrag [3] (Extractive Oracle Fragments) is an an extraction heuristic that "has access to the reference summary".

2.4.1 Abstractiveness of generated abstracts

[101] argued that state-of-the-art abstractive summarization systems that use a copy mechanism effectively generate the summary by copying over large chunks from the article, essentially doing "extractive" summarization. Following this work, we measure how much a model copies from the article by counting the proportion of n-grams from the generated abstract that are also found in the article. These statistics measured on the arXiv dataset are presented in figure 2.2. First, the original abstract and our TLM conditioned on the intro have small and very similar overlap fractions with the original article. A model using a pointing mechanism (we used our own implementation of the model developed by [2])[4] copies more than our transformer model, especially for higher n-grams. In particular, more than 10% of the 20-grams from the abstracts generated by the pointing model are also found in the article, showing that it tends to copy long sequences of words. On the other hand, our

[4] This model achieved the following ROUGE-1, 2, 3 and L on the arXiv dataset: $41.33, 14.73, 6.80, 36.34$

Model	Type	Extractive			Mixed			Abstractive		
		ROUGE								
		1	2	L	1	2	L	1	2	L
Previous Work										
Seq2Seq	Abs	6.1	0.2	5.4	5.7	0.2	5.1	6.2	1.1	5.7
TextRank	Ext	32.4	19.7	28.7	22.3	7.9	17.7	13.5	1.9	10.5
Pointer-gen	Mix	39.1	27.9	36.2	25.5	11.0	21.1	14.7	2.3	11.4
Lead-3	Ext	53.0	49.0	52.4	25.1	12.9	22.1	13.7	2.4	11.2
Exconsumm	Mix	**68.4**	**62.9**	**67.3**	31.7	16.1	27.0	17.1	3.1	14.1
Our Models										
Sent-CLF	Ext	53.0	47.0	52.1	26.8	12.6	23.6	15.4	2.7	12.8
Sent-PTR	Ext	60.7	55.2	59.7	28.9	14.1	25.1	15.9	2.8	13.0
TLM	Abs	49.8	39.7	47.4	27.1	11.6	22.8	**20.4**	**6.9**	**17.1**
TLM+E (G,M)	Mix	63.3	57.3	61.8	**31.9**	**16.6**	**27.4**	20.1	6.5	16.6
Oracle										
Gold Ext	Oracle	68.1	64.5	67.3	40.8	24.6	34.2	21.9	5.2	16.3
TLM+E (G,G)	Oracle	78.8	74.0	77.8	38.6	22.0	33.6	24.5	9.6	20.8

Table 2.5 Summarization results on the Newsroom dataset. Previous work results from [4] and [5]. Note that extractive/mixed/abstractive columns denote the type of ground-truth summary. The Newsroom dataset has targets that are extracted from the input (extractive), that are created with heuristics (mixed) and that are created by humans (abstractive). Also note that the "Type" column refers to the model type for each row.

proposed model produces more "abstractive" summaries, demonstrating its ability to paraphrase. Our model tends to copy longer sequences when conditioned on the introduction and the sentences from the extractor. We hypothesize that providing extracted sentences from the article that already contain a lot of words present in the reference abstract, makes the transformer's task easier, by allowing it to copy words and phrases from the extracted sentences. We find empirical evidence of this in figure 2.2, showing that the majority of n-gram copies come from the extracted sentences. For 5-grams, close to 2/3rd of the words copied are from the extracted sentences. As the number of grams increases to 25-grams, 4/5th of the words copied are from the extracted sentences.

2.4.2 T-SNE of learned word embeddings

We visualize the word embeddings learned by our TLM model using t-sne. We find that words that are often associated with computer science are clustered in a different part of space when compared to words associated with physics.

We use the arXiv REST API to find the submission category of each paper in the training

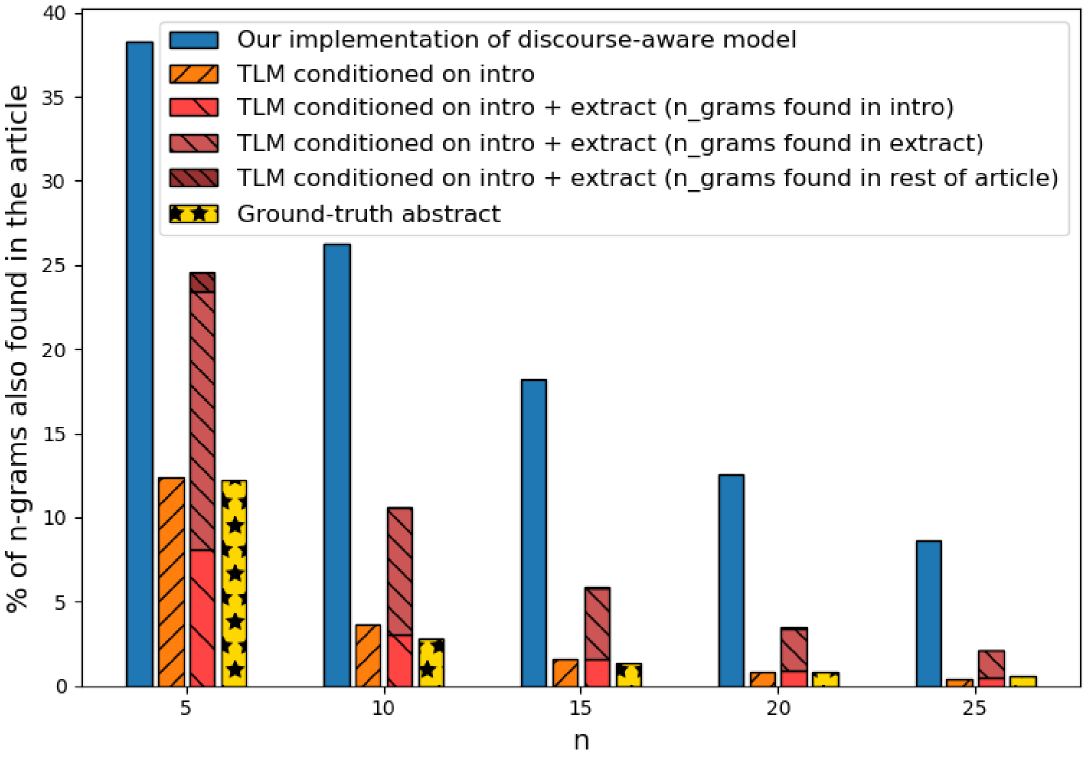


Figure 2.2 n-gram overlaps between the abstracts generated by different models and the input article on the arXiv dataset. We show in detail which part of the input was copied for our TLM conditioned on intro + extract.

set and then find the ∼300 most representative words for each category, using TF-IDF scores and plot them.

2.4.3 Human Evaluation

We performed a human evaluation using the same experimental setup as in [4] in Table 2.7. For the same 60 Newsroom test articles, we obtain the summaries for 5 different models (ground truth, sentence classifier, sentence pointer, TLM conditioned on article, TLM conditioned on article + pointer extracts). As expected, Transformers are quite good making coherent and fluent summaries but not necessarily on informativeness and relevance. Transformers have a logarithmic or constant path length (as opposed to linear in RNNs) between a network's output and any of its inputs, making gradient flow much easier. This is a clear advantage over RNNs that tend to repeat sentences. Transformers are also known to hallucinate [102] but we notice that including extracted sentences, TLM + Intro + Extract, improve relevance by 3% over TLM + Intro, bringing relevance closer to extractive methods.

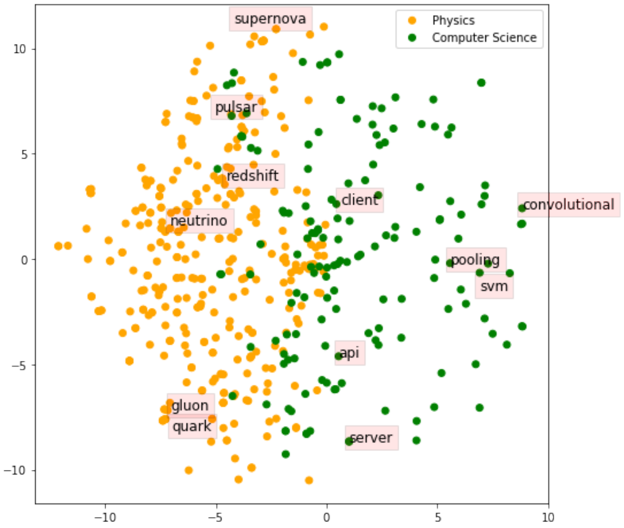


Figure 2.3 t-sne visualization of the TLM-learned word embeddings. The model appears to partition the space based on the broad paper categoty in which it frequently occurs.

Interestingly, on Coherence, both our TLM variants also score better than the ground truth. Over the four categories, TLM + Intro + Extract performs best on average over TLM + Intro, despite the former having higher ROUGE scores on the abstractive test set in table 2.5.

Somewhat counter-intuitively we observe that human written summaries are often rated lower than model summaries. However, other work has also found that human written ground truth summaries consistently receive lower scores when compared to model written summaries when evaluated by turkers (see for example Table 3 in the PEGASUS paper of [103]). We believe that this could be because Newsroom summaries are sometimes noisy, ungrammatical and incoherent.

2.4.4 Qualitative Results

Here we provide some qualitative results. Running our algorithm on a close to final version of this paper (excluding this section) and selecting the best sample from a set of 10-20 runs we found the following abstract: “we present a hybrid extractive and abstractive approach for generating summaries from long documents. we use an initial extractive step that reduces the amount of context for a subsequent abstractive step (see figure [fig: model]). we show that this approach can produce a good summarization quality on both short and long documents, even without using copying and pointing mechanisms. further, by considering the context in

Document — A new plan from the government of the Philippines would offer free wireless internet to people across the country while also likely eating into the annual revenue of the nations telecoms. Bloomberg reports that the Philippines government plans to roll-out its free Wi-Fi services to roughly half of the countrys municipalities over the next few months and the country has its sights set on nationwide coverage by the end of 2016. The free wireless internet service will be made available in public areas such as schools, hospitals, airports and parks, and is expected to cost the government roughly $32 million per year. [...]
Abstractive — : The government is reportedly considering a nationwide service plan to give free Wi-Fi access to rural areas.
Mixed — The government of the Philippines is considering a new plan to provide free wireless internet to the nation's largest cities and towns.
Extractive — The new plan will include free wireless internet to residents across the country while also probably eating into the annual revenue of the country's telecoms.
Document — (CBS) - Controversy over a new Microsoft patent has people questioning whether or not the intention has racist undertones. CNET reported that Microsoft has been granted a U.S. patent that will steer pedestrians away from areas that are high in crime. [...]
Absractive Summary — The new Microsoft patent claims a device could provide pedestrian navigation directions from a smartphone.
Mixed Summary Microsoft won a U.S. patent for a new way to steer pedestrians out of areas that are high in crime

Table 2.6 Qualitative Results - News articles and our model generated summaries on the NewsRoom dataset

both the text and the discourse, we find that the hybrid approach is effective at capturing the underlying context. we examine these models through rouge scores, through a study of the amount of n-gram copying performed by different models, as well as through a human evaluation using a standard protocol. our results show that our hybrid approach yields results that outperform current state-of-the-art results on several metrics of these evaluations."

Model	Type	Evaluation criteria			
		COH	FLU	INF	REL
Ground truth summaries	Orac	3.73	3.98	3.19	3.59
TLM - Intro + Extract	Mix	**3.78**	3.75	3.09	3.59
TLM - Intro	Mix	3.77	**3.90**	3.11	3.50
Sentence pointer	Ext	3.67	3.66	3.24	3.78
Sentence classifier	Ext	3.62	3.79	**3.47**	**3.89**

Table 2.7 Human evaluation on Newsroom abstractive summarization test data. Each pair of (article, summary) is presented to three unique crowd workers, who are asked to judge the summaries along four criteria: Coherence (COH: does the summary make sense as a whole), Fluency (FLU: is it well written), Informativeness (INF: does the summary catch the most important points of the article), and Relevance (REL: are the facts in the summary consistent with the article).

2.5 Conclusion

We have demonstrated that Transformer language models can generate high-quality summaries of long sequences of text via an extractive step followed by an abstractive step. We quantitatively measure the positive impact of the extractive step, by comparing it to a abstractive model variant that only sees the input text itself. Our approach outperforms previous extractive and abstractive summarization methods on the arXiv, PubMed and bigPatent datasets and is less prone to copying entire phrases or sentences from the input text. As with other problem domains, we have observed that abstractive summaries generated by transformers can generate imaginary content. We advise that such evaluations should probe multiple aspects of the summarization results including both factual correctness and coherency. We also note that for evaluating the correctness of the summaries of scientific articles and patents one must have highly trained evaluators who are willing to invest significant amounts of time to read the underlying papers and patents. Such studies could therefore require significant investments of resources. While we believe that this work is a step forward towards generating more abstractive summaries, it remains an open challenge to develop abstactive models that respect the underlying facts of the content being summarized while matching the creative ability of humans to coherently and concisely synthesize summaries. However, the TLMs we use here are of moderate size compared to what is now possible. It would be very interesting to see what kind of performance larger models could achieve.

Generated abstract for "the attention model paper" [104] - In this paper, we propose a neural machine translation model, which learns to align and translate a source sentence without encoding it into a fixed length vector. The proposed model allows the neural network to align and translate jointly, and in addition, predicts a target word based on the context vectors associated with these source positions. We show that the proposed model achieves significantly improved translation performance than the conventional encoder decoder neural network approach, when the sentences in the training corpus are long . Quantitative analysis reveals that the proposed model finds a linguistically plausible soft alignment between a source sentence and the target sentence. Finally, we compare the performance of the proposed model with the state-of-the-art standard neural network translation models and show that the proposed approach outperforms the conventional approach in terms of both translation performance and the quality of the translations.	**Original abstract** - Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder-decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.
Generated abstract for "the GPT-2 paper" [72] - Language models have been successful in creating rich representations of complex systems, including tasks such as sentiment analysis. In this paper we demonstrate their ability to generalize well across domains and tasks. We demonstrate the utility of language models by demonstrating that a variety of tasks can be modeled using language models. We demonstrate the potential of language models in downstream tasks by highlighting the ability of these models to perform a wide range of tasks in a zero shot setting.	**Original abstract** - Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

Table 2.8 Qualitative Results — Generated abstracts of select papers using our Intro Only TLM (part 1).

Generated abstract for "the BERT paper" [105] - We develop an architecture for deep bidirectional language models that allows the representation to fuse the left and the right context of a given token. Our framework uses bidirectional transformer encoder representations that enable the model to fuse the context representations of a token. We show that our architecture achieves state-of-the-art performance on a large suite of tasks, outperforming many systems with task-specific architectures. Our results are based on a set of eleven task specific tasks that are benchmarks for a large suite of nlp tasks.	**Original abstract** - We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).
Generated abstract for "the GAN paper" [106] - Adversarial nets are a class of deep generative networks that utilize the advantages of piecewise linear units. Here, we explore a novel model estimation procedure that combines several methods, such as backpropagation and dropout, to learn generative models. The proposed adversarial net framework can be thought of as analog to a team of counterfeiters, who try to produce fake currency and use it without being detected, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. We refer to this special case as adversarial nets. We explore the special case when the generative model generates samples by passing random noise through a multilayer perceptron, and the discriminative model is also a multilayer perceptron.	**Original abstract** - We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

Table 2.9 Qualitative Results — Generated abstracts of select papers using our Intro Only TLM (part 2).

CHAPTER 3 ARTICLE 2: CONDITIONALLY ADAPTIVE MULTI-TASK LEARNING: IMPROVING TRANSFER LEARNING IN NLP USING FEWER PARAMETERS & LESS DATA

Jonathan Pilault[1,*], Amine El hattami[1,*], Christopher Pal[1,2,3]

Accepted in International Conference on Learning Representations (ICLR), Publication Date: 01/12/2021

[1]Polytechnique Montreal & Mila [2]Element AI [3]Canada CIFAR AI Chair

{jonathan.pilault, amine.elhattami, christopher.pal}@polymtl.ca

* Joint first authors.

Abstract

Multi-Task Learning (MTL) networks have emerged as a promising method for transferring learned knowledge across different tasks. However, MTL must deal with challenges such as: overfitting to low resource tasks, catastrophic forgetting, and negative task transfer, or learning interference. Often, in Natural Language Processing (NLP), a separate model per task is needed to obtain the best performance. However, many fine-tuning approaches are both parameter inefficient, i.e., potentially involving one new model per task, and highly susceptible to losing knowledge acquired during pretraining. We propose a novel Transformer based Adapter consisting of a new conditional attention mechanism as well as a set of task-conditioned modules that facilitate weight sharing. Through this construction, we achieve more efficient parameter sharing and mitigate forgetting by keeping half of the weights of a pretrained model fixed. We also use a new multi-task data sampling strategy to mitigate the negative effects of data imbalance across tasks. Using this approach, we are able to surpass single task fine-tuning methods while being parameter and data efficient (using around 66% of the data for weight updates). Compared to other BERT Large methods on GLUE, our 8-task model surpasses other Adapter methods by 2.8% and our 24-task model outperforms by 0.7-1.0% models that use MTL and single task fine-tuning. We show that a larger variant of our single multi-task model approach performs competitively across 26 NLP tasks and yields state-of-the-art results on a number of test and development sets. Our code is publicly available at `https://github.com/CAMTL/CA-MTL`[1].

[1] Jonathan Pilault a contribué dans la conception des algorithmes, l'experimentation, l'analyse des résultats, amélioration de l'efficacité du logiciel, analyse théorique, rédaction de l'article, recherche bibliographique

3.1 Introduction

The introduction of deep, contextualized Masked Language Models (MLM)[2] trained on massive amounts of unlabeled data has led to significant advances across many different Natural Language Processing (NLP) tasks [107, 108]. Much of these recent advances can be attributed to the now well-known BERT approach [6]. Substantial improvements over previous state-of-the-art results on the GLUE benchmark [16] have been obtained by multiple groups using BERT models with task specific fine-tuning. The "BERT-variant + fine-tuning" formula has continued to improve over time with newer work constantly pushing the state-of-the-art forward on the GLUE benchmark. The use of a single neural architecture for multiple NLP tasks has shown promise long before the current wave of BERT inspired methods [109] and recent work has argued that autoregressive language models (ARLMs) trained on large-scale datasets – such as the GPT family of models [110], are in practice multi-task learners [111]. However, even with MLMs and ARLMs trained for multi-tasking, single task fine-tuning is usually also employed to achieve state-of-the-art performance on specific tasks of interest. Typically this fine-tuning process may entail: creating a task-specific fine-tuned model [6], training specialized model components for task-specific predictions [8] or fine-tuning a single multi-task architecture [9].

Single-task fine-tuning overall pretrained model parameters may have other issues. Recent analyses of such MLM have shed light on the linguistic knowledge that is captured in the hidden states and attention maps [112, 113, 114]. Particularly, BERT has middle Transformer [115] layers that are typically the most transferable to a downstream task [108]. The model proxies the steps of the traditional NLP pipeline in a localizable way [113] — with basic syntactic information appearing earlier in the network, while high-level semantic information appearing in higher-level layers. Since pretraining is usually done on large-scale datasets, it may be useful, for a variety of downstream tasks, to conserve that knowledge. However, single task fine-tuning cause catastrophic forgetting of the knowledge learned during MLM [116]. To preserve knowledge, freezing part of a pretrained network and using *Adapters* for new tasks have shown promising results [8].

Inspired by the human ability to transfer learned knowledge from one task to another new task, Multi-Task Learning (MTL) in a general sense [117, 118, 119] has been applied in

et la réponse aux réviseurs de conférence.

[2] For reader convenience, all acronyms in this paper are summarized in section B.1.1 of the Appendix.

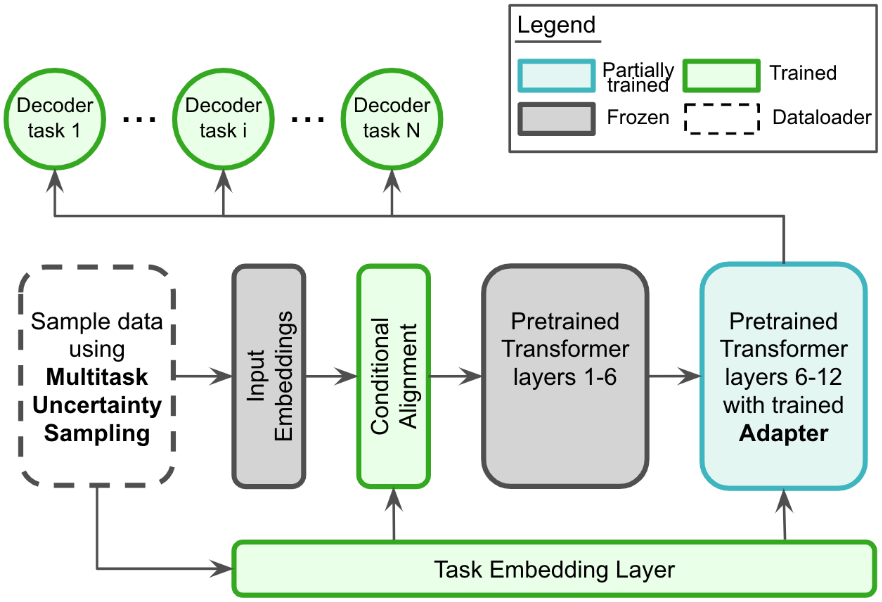


Figure 3.1 CA-MTL base architecture with our uncertainty-based sampling algorithm. Each task has its own decoder. The input embedding layer and the lower Transformer layers are frozen. The upper Transformer layer and Conditional Alignment module are modulated with the task embedding.

many fields outside of NLP. [120] showed that a model trained in a *multi-task* manner can take advantage of the inductive transfer between tasks, achieving a better generalization performance. MTL has the advantage of computational/storage efficiency [121], but training models in a multi-task setting is a balancing act; particularly with datasets that have different: **(a)** dataset sizes, **(b)** task difficulty levels, and **(c)** different types of loss functions. In practice, learning multiple tasks at once is challenging since negative transfer [122], task interference [123, 124] and catastrophic forgetting [125] can lead to worse data efficiency, training stability and generalization compared to single task fine-tuning.

Using Conditionally Adaptive Learning, we seek to improve pretraining knowledge retention and multi-task inductive knowledge transfer. Our contributions are the following:

- A new task conditioned Transformer that adapts and modulates pretrained weights **(Section 3.2.1)**.
- A novel way to prioritize tasks with an uncertainty based multi-task data sampling method that helps balance the sampling of tasks to avoid catastrophic forgetting **(Section 3.2.2)**.

Our Conditionally Adaptive Multi-Task Learning (CA-MTL) approach is illustrated in Figure 3.1. To the best of our knowledge, our work is the first to explore the use of a latent representation of tasks to modularize and adapt pretrained architectures. Further, we believe our work is also the first to examine uncertainty sampling for large-scale multi-task learning in

NLP. We show the efficacy of CA-MTL by: **(a)** testing on 26 different tasks and **(b)** presenting state-of-the-art results on a number of test sets as well as superior performance against both single-task and MTL baselines. Moreover, we further demonstrate that our method has advantages over **(c)** other adapter networks, and **(d)** other MTL sampling methods. Finally, we provide ablations and separate analysis of the MT-Uncertainty Sampling technique in section 3.4.1 and of each component of the adapter in 3.4.2.

3.2 Methodology

This section is organized according to the two main MTL problems that we will tackle: (1) How to modularize a pretrained network with latent task representations? (2) How to balance different tasks in MTL? We define each task as: $\mathscr{T}_i \triangleq \{p_i(\mathbf{y}_i|\mathbf{x}_i, \mathbf{z}_i), \mathscr{L}_i, \tilde{p}_i(\mathbf{x}_i)\}$, where $\mathbf{z}_i$ is task i's learnable shallow embedding, $\mathscr{L}_i$ is the task loss, and $\tilde{p}_i(\mathbf{x}_i)$ is the empirical distribution of the training data pair $\{\mathbf{x}_i, \mathbf{y}_i\}$, for $i \in \{1, \ldots, T\}$ and T the number of supervised tasks. The MTL objective is:

$$\min_{\phi(\mathbf{z}),\theta_1,\ldots,\theta_T} \sum_{i=1}^{T} \mathscr{L}_i(f_{\phi(\mathbf{z}_i),\theta_i}(\mathbf{x}_i), \mathbf{y}_i) \tag{3.1}$$

where f is the predictor function (includes encoder model and decoder heads), $\phi(\mathbf{z})$ are learnable generated weights conditioned on $\mathbf{z}$, and θ_i are task-specific parameters for the output decoder heads. $\mathbf{z}$ is constructed using an embedding lookup table.

3.2.1 Task Conditioned Transformer

Our task conditioned Transformer architecture is based on one simple concept. We either add conditional layers or modulate existing pretrained weights using a task representation by extending Feature Wise Linear Modulation [126] functions in several ways depending on the Transformer layer. We define our framework below.

Definition 1 (Conditional Weight Transformations). Given a neural network weight matrix $\mathbf{W}$, we compute transformations of the form $\phi(\mathbf{W}|\mathbf{z}_i) = \gamma_i(\mathbf{z}_i)\mathbf{W} + \beta_i(\mathbf{z}_i)$, where γ_i and β_i are learned functions that transform the weights based on a learned vector embedding $\mathbf{z}_i$, for task i.

Definition 2 (Conditionally Adaptive Learning). In our setting, Conditionally Adaptive Learning is the process of learning a set of ϕs for the conditionally adaptive modules presented below along with a set of task embedding vectors $\mathbf{z}_i$ for T tasks, using a multi-task loss (see equation 3.1).

In the subsections that follow: We introduce a new Transformer Attention Module using block-diagonal Conditional Attention that allows the original query-key based attention to account for task-specific biases (section **3.2.1**). We propose a new Conditional Alignment method that aligns the data of diverse tasks and that performs better than its unconditioned and higher capacity predecessor (section **3.2.1**). We adapt layer normalization statistics to specific tasks using a new Conditional Layer Normalization module (section **3.2.1**). We add a Conditional Bottleneck that facilitates weight sharing and task-specific information flow from lower layers (section **3.2.1**). In our experiments we provide an ablation study of these components (Table 3.1) examining performance in terms of GLUE scores.

Conditional Attention

Given d, the input dimensions, the query $\mathbf{Q}$, the key $\mathbf{K}$, and the value $\mathbf{V}$ as defined in [115], we redefine the attention operation:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{z}_i)) = \text{softmax}\left[M(\mathbf{z}_i) + \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right]\mathbf{V}$$

$$M(\mathbf{z}_i) = \bigoplus_{n=1}^{N} A'_n(\mathbf{z}_i) = \text{diag}(A'_1, \dots, A'_N), \quad A'_n(\mathbf{z}_i) = A_n \gamma_i(\mathbf{z}_i) + \beta_i(\mathbf{z}_i)$$

where $\bigoplus$ is the direct sum operator (see section 3.2.1), N is the number of block matrices $A_n \in \mathbb{R}^{(L/N)\times(L/N)}$ along the diagonal of the attention matrix, L is the input sequence, $M(\mathbf{z}_i) = \text{diag}(A'_1, \dots, A'_N)$ is a block diagonal conditional matrix. Note that A_n is constructed using L/N trainable and randomly initialized L/N dimensional vectors. While the original attention matrix depends on the hidden states h, $M(\mathbf{z}_i)$ is a learnable weight matrix that only depends on the task embedding $\mathbf{z}_i \in \mathbb{R}^d$. $\gamma_i, \beta_i : \mathbb{R}^d \mapsto \mathbb{R}^{L^2/N^2}$ are Feature Wise Linear Modulation [126] functions. We also experimented with full-block Conditional Attention $\in \mathbb{R}^{L\times L}$. Not only did it have N^2 more parameters compared to the block-diagonal variant, but it also performed significantly worse on the GLUE development set (see FBA variant in Table B.4). It is possible that GLUE tasks derive a certain benefit from localized attention that is a consequence of $M(\mathbf{z}_i)$. With $M(\mathbf{z}_i)$, each element in a sequence can only attend to other elements in its subsequence of length L/N. In our experiments we used $N = d/L$. The full Conditional Attention mechanism used in our experiments is illustrated in Figure 3.2.

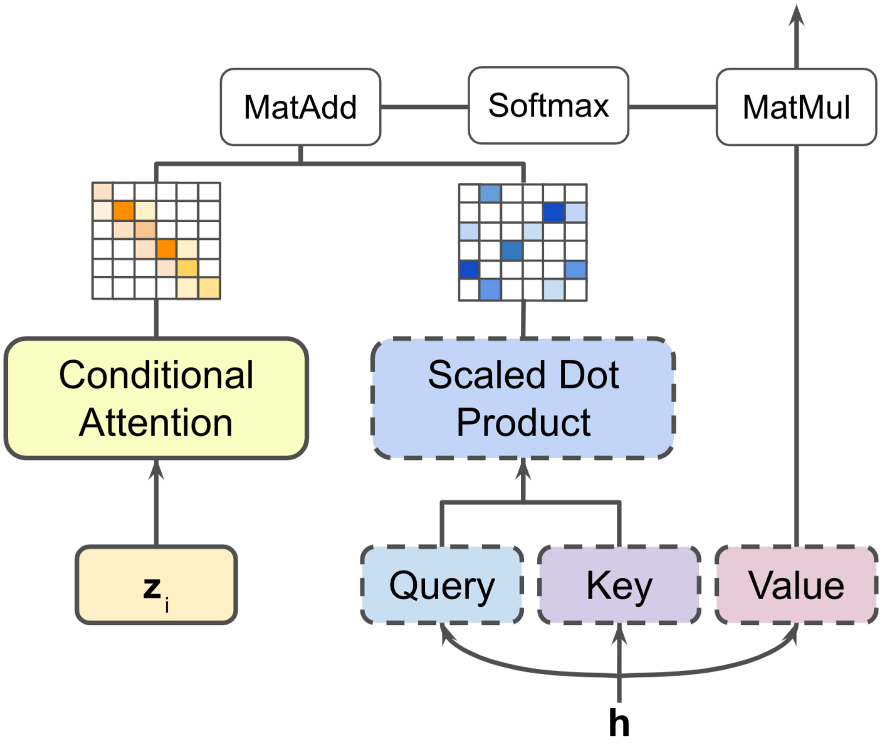


Figure 3.2 Conditional Attention Module

The Direct Sum Operator

In section 3.2.1, we used the direct sum operator $\oplus$. This operation allows us to create a block diagonal matrix. The direct sum of a matrix $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{p\times q}$ results in a matrix of size $(m+p)\times(n+q)$, defined as:

$$\mathbf{A}\oplus\mathbf{B} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & b_{11} & \cdots & b_{1q} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & b_{p1} & \cdots & b_{pq} \end{bmatrix}$$

Conditional Alignment

[123] showed that in MTL having T separate alignment modules $R_1, \ldots, R_T$ increases BERT$_{\text{LARGE}}$ avg. scores on five GLUE tasks (CoLA, MRPC, QNLI, RTE, SST-2) by 2.35%. Inspired by this work, we found that adding a task conditioned alignment layer between the input embedding layer and the first BERT Transformer layer improved multi-task model perfor-

mance. However, instead of having T separate alignment matrices R_i for each T task, one alignment matrix $\hat{R}$ is generated as a function of the task embedding z_i. As in [123], we tested this module on the same five GLUE tasks and with BERT$_{\text{LARGE}}$. Enabling task conditioned weight sharing across covariance alignment modules allow us to outperforms BERT$_{\text{LARGE}}$ by 3.61%. This is 1.26 % higher than having T separate alignment matrices. Inserting $\hat{R}$ into BERT, yields the following encoder function $\hat{f}$:

$$\hat{f} = \sum_{t=1}^{T} g_{\theta_i}(E(\mathbf{x}_i)\hat{R}(\mathbf{z}_i)B), \qquad \hat{R}(\mathbf{z}_i) = R\gamma_i(\mathbf{z}_i) + \beta_i(\mathbf{z}_i) \tag{3.2}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the layer input, g_{θ_i} is the decoder head function for task i with weights θ_i, E the frozen BERT embedding layer, B the BERT Transformer layers and R the linear weight matrix of a single task conditioned alignment matrix. $\gamma_i, \beta_i : \mathbb{R}^d \mapsto \mathbb{R}^d$ are Feature Wise Linear Modulation functions.

Conditional Bottleneck

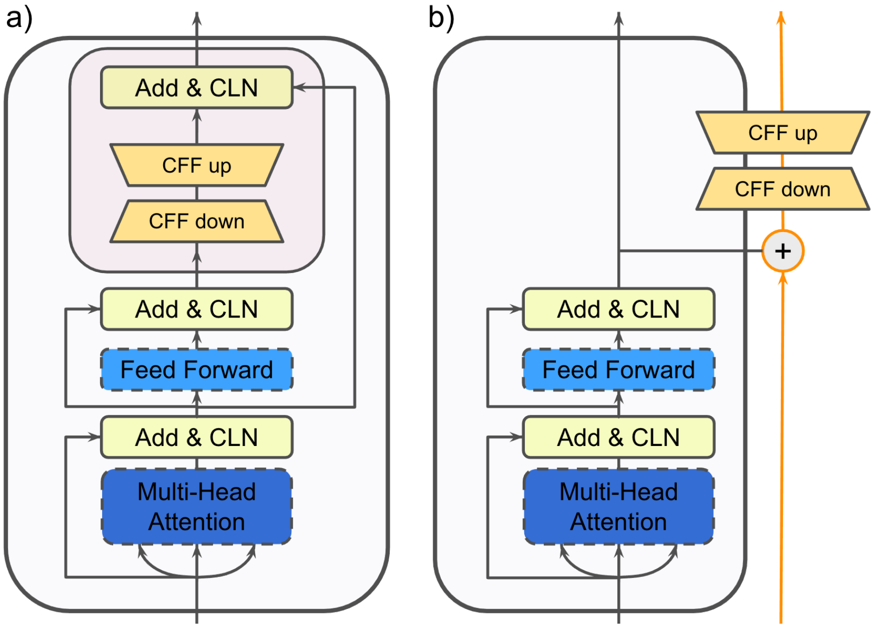


Figure 3.3 a) Conditional Bottleneck for CA-MTL$_{\text{BASE}}$. b) Conditional Bottleneck for CA-MTL$_{\text{LARGE}}$.

We created a task conditioned two layer feed-forward bottleneck layer (CFF up/down in Figure 3.3). The conditional bottleneck layer follows the same transformation as in equation 3.2. The module in Figure 3.3a is added to the top most Transformer layers of CA-MTL$_{\text{BASE}}$ and uses a CLN. For CA-MTL$_{\text{LARGE}}$ this module is the main building block of the skip connection added alongside all Transformer layers seen in Figure 3.3b. The connection at layer j takes in the matrix sum of the Transformer layer output at j and the previous connection's output at $j-1$. The Conditional bottleneck allows lower layer information to flow upwards depending on the task. Our intuition for introducing this component is related to recent studies [113] that showed that the "most important layers for a given task appear at specific positions". As with the other modules described so far, each task adaptation is created from the weights of a single shared adapter that is modulated by the task embedding.

Conditional Layer Normalization (CLN)

We extend the Conditional Batch Normalization idea from [127] to Layer Normalization [128]. For task $\mathscr{T}_i$, $i \in \{1, \ldots, T\}$:

$$\mathbf{h}_i = \frac{1}{\sigma} \odot (\mathbf{a}_i - \mu) * \hat{\gamma}_i(\mathbf{z}_i) + \beta_i(\mathbf{z}_i), \qquad \hat{\gamma}_i(\mathbf{z}_i) = \boldsymbol{\gamma}'\gamma_i(\mathbf{z}_i) + \boldsymbol{\beta}' \tag{3.3}$$

where $\mathbf{h}_i$ is the CLN output vector, $\mathbf{a}_i$ are the preceding layer activations associated with task i, μ and σ are the mean and the variance of the summed inputs within each layer as defined in [128]. Conditional Layer Normalization is initialized with BERT's Layer Normalization affine transformation weights and bias $\boldsymbol{\gamma}'$ and $\boldsymbol{\beta}'$ from the original formulation: $\mathbf{h} = \frac{1}{\sigma} \odot (\mathbf{a} - \mu) * \boldsymbol{\gamma}' + \boldsymbol{\beta}'$. During training, the weight and bias functions of $\gamma_i(*)$ and $\beta_i(*)$ are always trained, while the original Layer Normalization weight may be kept fixed. This module was added to account for task specific rescaling of individual training cases. Layer Normalization normalizes the inputs across features. The conditioning introduced in equation 3.2.1 allows us to modulate the normalization's output based on a task's latent representation.

3.2.2 Multi-Task Uncertainty Sampling

MT-Uncertainty Sampling is a task selection strategy that is inspired by Active Learning techniques. Our algorithm 1 is outlined below. Similar to Active Learning, our algorithm first evaluates the model uncertainty. MT-Uncertainty Sampling uses Shannon Entropy, an uncertainty measure, to choose training examples by first doing forward pass through the model with $b \times T$ input samples. For an output classification prediction with C_i possible classes and probabilities $(p_{i,1}, \ldots, p_{i,C_i})$, the Shannon Entropy H_i, for task $\mathscr{T}_i$ and $i \in \{1, \ldots, T\}$, our uncertainty measure $\mathscr{U}(\mathrm{x})$ are given by:

$$H_i = H_i(f_{\phi(\mathbf{z}_i),\theta_i}(\mathrm{x})) = -\sum_{c=1}^{C_i} p_c \log p_c, \qquad \mathscr{U}(x_i) = \frac{H_i(f_{\phi(\mathbf{z}_i),\theta_i}(\mathrm{x}))}{\hat{H} \times H_i'} \tag{3.4}$$

$$\hat{H} = \max_{i \in \{1,\ldots,T\}} \bar{H}_i = \max \left[\frac{1}{b} \sum_{\mathrm{x} \in \mathbf{x}_i} H_i \right], \qquad H_i' = -\sum_{c=1}^{C_i} \frac{1}{C_i} \log \left[\frac{1}{C_i} \right] \tag{3.5}$$

where $\bar{H}_i$ is the average Shannon Entropy across b samples of task t, H_i', the Shannon entropy of choosing classes with uniform distribution and $\hat{H}$, the maximum of each task's average entropy over b samples. H_i' is normalizing factor that accounts for differing number of prediction classes (without the normalizing factor H_i', tasks with a binary classification $C_i = 1$ were rarely chosen). Further, to limit high entropy outliers and to favor tasks with highest uncertainty, we normalize with $\hat{H}$. The measure in eq. 3.4 allows Algorithm 1 to choose b samples from $b \times T$ candidates to train the model.

```
Input: Training data $D_t$ for task $t \in [1, \ldots, T]$; batch size $b$; $C_t$ possible
       output classes for task $t$; $f := f_{\phi(\mathbf{z}_i),\theta_i}$ our model with weights $\phi, \theta_i$;
Output: $\mathcal{B}'$ - multi-task batch of size $b$
$\mathcal{B} \leftarrow \emptyset$
for $t \leftarrow 1$ to $T$ do
    Generate $\mathbf{x}_t := \{x_{t,1}, \ldots, x_{t,b}\} \overset{\text{i.i.d.}}{\sim} D_t$
    for $i \leftarrow 1$ to $b$ do
        $\mathcal{H}_{t,i} \leftarrow -\sum_{c=1}^{C_i} p_c(f(x_{t,i})) \log p_c(f(x_{t,i}))$        ▷ Entropy of each sample
    end
    Compute $\bar{\mathcal{H}}_t \leftarrow \frac{1}{b} \sum_{\mathrm{x} \in \mathbf{x}_i} \mathcal{H}_{t,i}$        ▷ Average entropy for task $t$
    Compute $H'_t \leftarrow -\sum_{c=1}^{C_t} \frac{1}{C_t} \log \left[\frac{1}{C_t}\right]$        ▷ Max entropy (uniform distribution)
    $\mathcal{B} \leftarrow \mathcal{B} \cup \mathbf{x}_t$ and $D_t \leftarrow D_t \setminus \mathbf{x}_t$
    if $D_t = \emptyset$ then
        Reload $D_t$
    end
    for $i \leftarrow 1$ to $b$ do
        Compute: $\mathcal{U}_{t,i} \leftarrow \mathcal{H}_{t,i} / H'_t$        ▷ Uncertainty normalized with max entropy
    end
end
Compute $\hat{\mathcal{H}} \leftarrow \max_{i \in \{1,\ldots,T\}} [\bar{\mathcal{H}}_t]$        ▷ Entropy of task with highest average entropy
Update $\mathcal{U}_{t,i} \leftarrow \mathcal{U}_{t,i} / \hat{\mathcal{H}}$        ▷ Normalize each sample's uncertainty measure
$\mathcal{B}' \leftarrow \mathrm{top_b}(\{\mathcal{U}_{t,i} | t \in [1, \ldots, T], i \in [1, \ldots, b]\})$        ▷ $b$ samples w/ highest uncertainty
Return: With $\mathcal{B}'$, solve eq. 3.1 with gradient descent; updated model $f$
```

Algorithm 1: Multi-task Uncertainty Sampling

An advantage of our MT-Uncertainty Sampling approach is its ability to manage task difficulty. This is highlighted in Figure 3.4. In this experiment, we estimated task difficulty using the Evolutionary Data Measures (EDM)[3] proposed by [129]. The task difficulty estimate relies on multiple dataset statistics such as the data size, class diversity, class balance and class interference. Interestingly, estimated task difficulty correlates with the first instance that the selection of a specific task occurs. Supposing that QNLI is an outlier, we notice that peaks in the data occur whenever tasks are first selected by MT Uncertainty sampling. This process follows the following order: 1. MNLI 2. CoLA 3. RTE 4. QQP 5. MRPC 6.SST-2, which is the order from highest task difficulty to lowest task difficulty using EDM. As opposed to Curriculum Learning [130], MT-Uncertainty dynamically prioritizes the most difficult tasks. As also discovered in MTL vision work [131], this type of prioritization on more difficult tasks may explain MT-Uncertainty's improved performance over other task selection methods. In MTL, heuristics to balance tasks during training is typically done by weighting each task's loss differently. We see here how MT-Uncertainty is able to prioritize task difficulty.

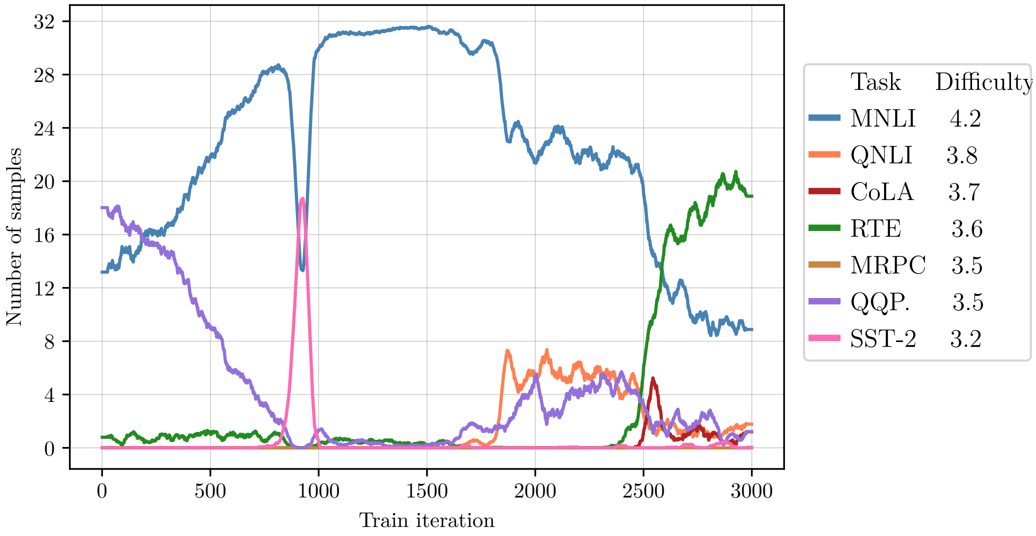


Figure 3.4 Task composition of MT-Uncertainty sampling and estimated task difficulty using EDM: number of training samples per task at each iteration for batch size of 32. The occurrence of first peaks and estimated difficulty follow the same order: From highest to lowest: MNLI > CoLA > RTE > QQP = MRPC > SST-2.

While the EDM difficulty measure is shown to correlate well with model performance, it lacks precision. As reported in [129], the average score achieved on the Yahoo Answers dataset is

[3] https://github.com/Wluper/edm

69.9% and its difficulty is 4.51. The average score achieved on Yelp Full is 56.8%, 13.1% less than Yahoo Answers and its difficulty is 4.42. The authors mention that “This indicates that the difficulty measure in its current incarnation may be more effective at assigning a class of difficulty to datasets, rather than a regression-like value”.

3.3 Related Work

Multi-Tasking in NLP. To take advantage of the potential positive transfer of knowledge from one task to another, several works have proposed carefully choosing which tasks to train as an intermediate step in NLP before single task fine-tuning [132, 133, 122, 134, 135, 14]. The intermediate tasks are not required to perform well and are not typically evaluated jointly. In this work, all tasks are trained *jointly* and *all tasks used* are evaluated from a *single model*. In Natural Language Understanding (NLU), it is still the case that to get the best task performance one often needs a separate model per task [136, 137]. At scale, Multilingual NMT systems [138] have also found that MTL model performance degrades as the number of tasks increases. We notice a similar trend in NLU with our baseline MTL model. Recently, approaches in MTL have tackled the problem by designing task specific decoders on top of a shared model [9] or distilling multiple single-task models into one [136]. Nonetheless, such MTL approaches still involves single task fine-tuning. In this paper, we show that it is possible to achieve high performance in NLU without single task fine-tuning.

Adapters. Adapters are trainable modules that are attached in specific locations of a pretrained network. They provide another promising avenue to limit the number of parameters needed when confronted with a large number of tasks. This approach is useful with pretrained MLM models that have rich linguistic information [139, 112, 108, 113]. Recently, [8] added an adapter to a pretrained BERT model by fine-tuning the layer norms and adding feed forward bottlenecks in every Transformer layer. However, such methods adapt each task individually during the fine-tuning process. Unlike prior work, our method harnesses the vectorized representations of tasks to modularize a single pretrained model across all tasks. [7] and [140] also mix both MTL and adapters with BERT and T5 encoder-decoder [141] respectively by creating local task modules that are controlled by a global task agnostic module. The main drawback is that a new set of non-shared parameters must be added when a new task is introduced. CA-MTL shares all parameters and is able to re-modulate existing weights with a new task embedding vector.

Multi-Tasking in NLP and other fields. MTL weight sharing algorithms such as Mixture-of-Experts (MoE) have found success in NLP [142]. CA-MTL can complement MoE since the Transformers multi-headed attention can be seen as a form of MoE [143]. In Vision, MTL can also improve with optimization [144] or gradient-based approaches [145, 124].

Active Learning, Task Selection and Sampling. [146] examined multi-task active learning for neural semantic role labeling in a low resource setting, using entity recognition as the sole auxiliary task. They used uncertainty sampling for active learning and found that 12% less data could be used compared to passive learning. [147] has examined different active learning techniques for the two task annotation scenario, focusing on named entity recognition and syntactic parse tree annotations. In contrast, here we examine the larger scale data regime, the modularization of a multi-task neural architecture, and the many task ($\gg$ 2) setting among other differences. Other than MTAL [147, 146], [148] leveraged model uncertainty to balance MTL losses but not to select tasks as is proposed here. Our sampling technique is similar to the ones found in several active learning algorithms [149] that are based on Shannon entropy estimations. [147] and [146] examined Multi-Task Active Learning (MTAL), a technique that chooses one informative sample for T different learners (or models) for each T tasks. Instead we choose T tasks samples for *one model.* Moreover, the algorithm weights each sample by the corresponding task score, and the Shannon entropy is normalized to account for various losses (see equation 3.5). Also, our algorithm is used in a large scale MTL setup ($\gg$ 2 tasks). Recently, [39] explored task selection in MTL using learning policies based on counterfactual estimations [150]. However, such method considers only fixed stochastic parameterized policies while our method *adapts* its selection criterion based on model uncertainty throughout the training process.

Hypernetworks. CA-MTL is a hypernetwork adapter. The method to generate *task-conditioned* adapter weights is inspired by [151]. Hypernetwork layers have also been fine-tuned along with pretrained models. For example, [152] uses stochastic variational inference [153] to produce language and task latent codes that conditionally generates the weights of a BERT prediction head, a single hypernetwork linear layer shared across multiple languages and tasks. Unlike previous methods however, CA-MTL conditionally modulates pretrained weights and biases, attention matrices, hidden representations and normalization statistics with task embeddings. Further, CA-MTL can preserve the pretraining knowledge by freezing the underlying Transformer model. Finally, we show a synergy between our hypernetwork adapter and our active task sampling technique (see section 3.2.2) that allows CA-MTL to continue surpassing fully tuned models as we scale the number of tasks (see figure 3.8).

3.4 Experiments and Results

We show that our adapter of section 3.2 achieves parameter efficient transfer for 26 NLP tasks. Our implementation of CA-MTL is based on HuggingFace [154]. Hyperparameters and our experimental set-up are outlined in B.1.3. To preserve the weights of the pretrained model, CA-MTL's bottom half Transformer layers are frozen in all experiments (except in section 3.4.4). We also tested different layer freezing configurations and found that freezing half the layers worked best on average (see Section B.1.5).

3.4.1 Multi-Task Uncertainty Sampling

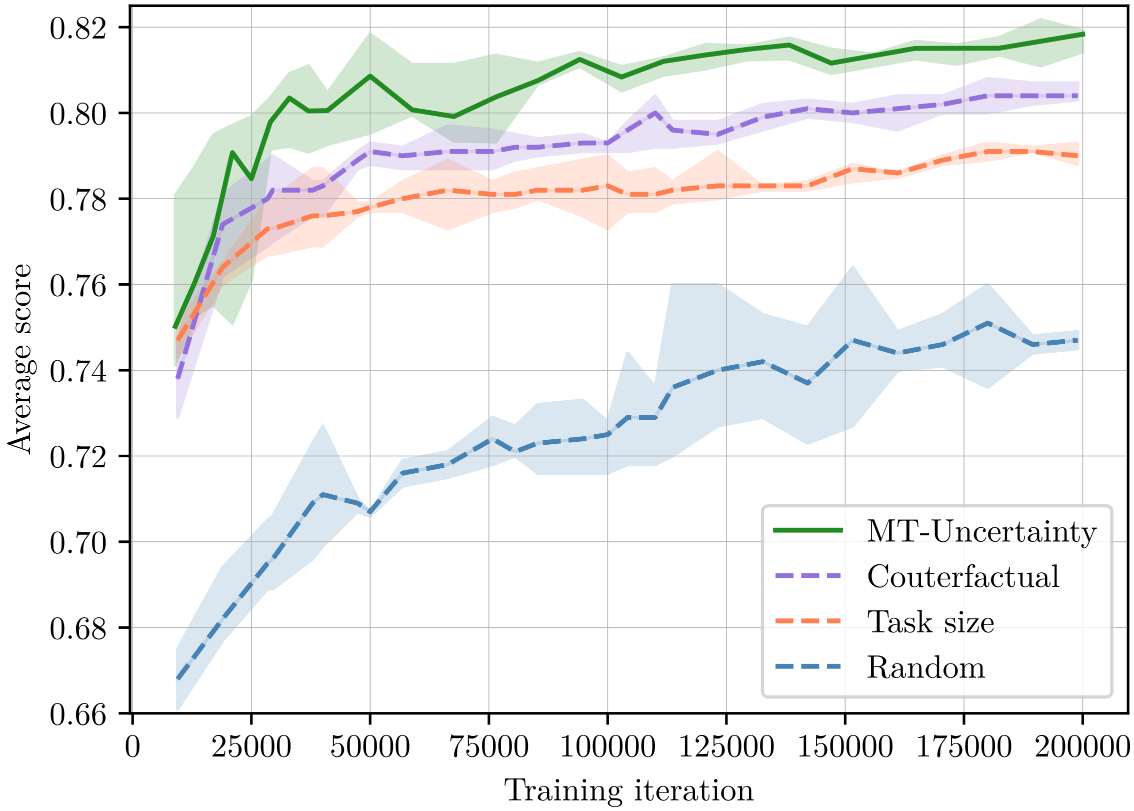
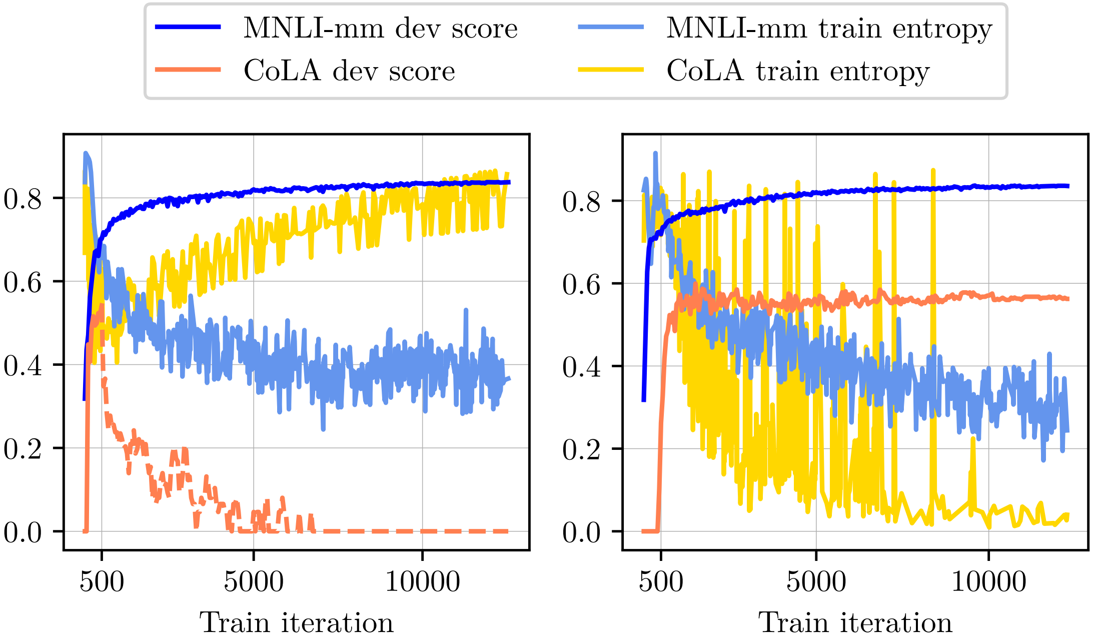


Figure 3.5 **MT-Uncertainty** vs. other task sampling strategies: median dev set scores on 8 GLUE tasks using $\text{BERT}_{\text{BASE}}$. Data for the Counterfactual and task-size policy $\pi_{|\text{task}|}$ (Eq. 3.6) from [39].

Our MT-Uncertainty sampling strategy, from section 3.2.2, is compared to 3 other task selection schemes: a) Counterfactual b) Task size c) Random. We used a $\text{BERT}_{\text{BASE}}$ (no adapters) on 200k iterations and with the same hyperparameters as in [39]. For more information on Counterfactual task selection, we invite the reader to consult the full explanation in [39]. For T tasks and the dataset D_i for tasks $i \in \{1, \ldots, T\}$, we rewrite the definitions of Random π_{rand} and Task size $\pi_{|task|}$ sampling:

$$\pi_{rand} = 1/T, \quad \pi_{|task|} = |D_i| \left[\sum_{i=1}^{T} |D_i| \right]^{-1} \tag{3.6}$$

In Figure 3.5, we see from the results that MT-Uncertainty converges faster by reaching the 80% average GLUE score line before other task sampling methods. Further, MT-Uncertainty maximum score on 200k iterations is at 82.2, which is 1.7% higher than Counterfactual sampling. The datasets in the GLUE benchmark offers a wide range of dataset sizes. This is useful to test how MT-Uncertainty manages a jointly trained low resource task (CoLA) and high resource task (MNLI). Figure 3.6 explains how catastrophic forgetting is curtailed by sampling tasks before performance drops. With π_{rand}, all of CoLA's tasks are sampled by iteration 500, at which point the larger MNLI dataset overtakes the learning process and CoLA's dev set performance starts to diminish. On the other hand, with MT-Uncertainty sampling, CoLA is sampled whenever Shannon entropy is higher than MNLI's. The model first assesses uncertain samples using Shannon Entropy then decides what data is necessary to train on. This process allows lower resource tasks to keep performance steady. We provide evidence in Figure 3.4 that MT-Uncertainty is able to manage task difficulty — by choosing the most difficult tasks first.



(a) Random

(b) MT-Uncertainty

Figure 3.6 CoLA/MNLI dev set scores and entropy for π_{rand} (left) and **MT-Uncertainty** (right).

3.4.2 Ablation and Module Analysis

In Table 3.1, we present the results of an ablation study to determine which elements of CA-MTL$_{\text{BERT-BASE}}$ had the largest positive gain on average GLUE scores. Starting from a MTL BERT$_{\text{BASE}}$ baseline trained using random task sampling (π_{rand}). Apart for the Conditional Adapter, each module as well as MT-Uncertainty lift overall performance and reduce variance across tasks. Please note that we also included accuracy/F1 scores for QQP, MRPC and Pearson/ Spearman correlation for STS-B to calculate score standard deviation Task σ. Intuitively, when negative task transfer occurs between two tasks, either (1) task interference is bidirectional and scores are both impacted, or (2) interference is unidirectional and only one score is impacted.

Table 3.1 Model ablation study[a] on the GLUE dev set. All models have the bottom half layers frozen.

Model changes	Avg GLUE	Task σ GLUE	% data used
BERT$_{\text{BASE}}$ MTL (π_{rand})	80.61	14.41	100
+ Conditional Attention	82.41	10.67	100
+ Conditional Adapter	82.90	11.27	100
+ CA and CLN	83.12	10.91	100
+ MT-Uncertainty (CA-MTL$_{\text{BERT-BASE}}$)	**84.03**	**10.02**	66.3

[a] CA = Conditional Alignment, CLN = Conditional Layer Normalization, Task σ = scores standard deviation *across tasks*.

We calculate Task σ to characterize changes in the dynamic range of performance across multiple tasks. We do this to asses the degree to which performance improvements are distributed across all tasks or only subsets of tasks. As we can see from Table 3.1, Conditional Attention, Conditional Alignment, Conditional Layer Normalization, MT-Uncertainty play roles in reducing Task σ and increasing performance across tasks. This provides partial evidence of CA-MTL's ability to mitigating negative task transfer.

We show that Conditional Alignment can learn to capture covariate distribution differences with task embeddings co-learned from other adapter components of CA-MTL. In Figure 3.7, we arrive at similar conclusions as [123], who proved that negative task transfer is reduced

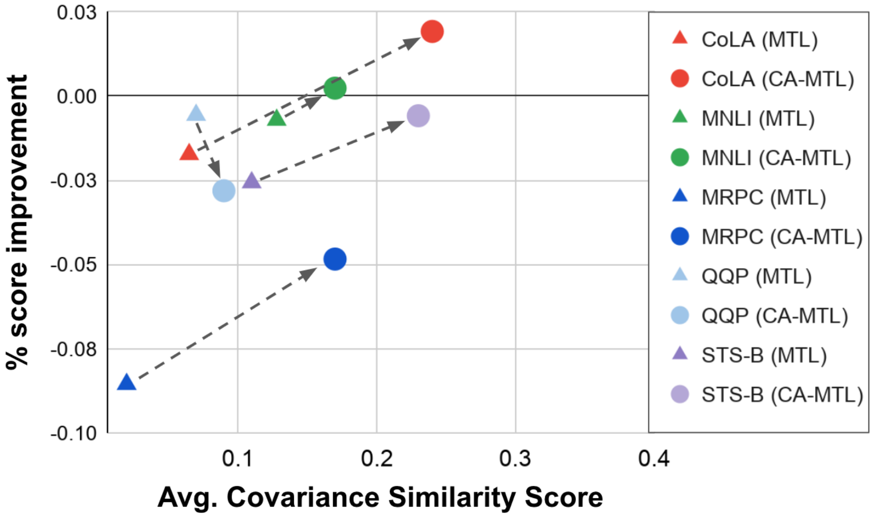


Figure 3.7 Task performance vs. avg. covariance similarity scores (eq. 3.7) for MTL and CA-MTL.

when task covariances are aligned. The authors provided a "covariance similarity score" to gauge covariance alignment. For task i and j with m_i and m_j data samples respectively, and given d dimensional inputs to the first Transformer layer $X_i \in \mathbb{R}^{m_i \times d}$ and $X_j \in \mathbb{R}^{m_j \times d}$, we rewrite the steps to calculate the covariance similarity score between task i and j: (a) Take the covariance matrix $X_i^\top X_i$, (b) Find its best rank-r_i approximation $U_{i,r_i} D_{i,r_i} U_{i,r_i}^\top$, where r_i is chosen to contain 99% of the singular values. (c) Apply steps (a), (b) to X_j, and compute the covariance similarity score $CovSim_{i,j}$:

$$CovSim_{i,j} := \frac{\|(U_{i,r_i} D_{i,r_i}^{1/2})^\top U_{j,r_j} D_{j,r_j}^{1/2}\|_F}{\|U_{i,r_i} D_{i,r_i}^{1/2}\|_F \cdot \|U_{j,r_j} D_{j,r_j}^{1/2}\|_F}. \quad CovSim_i = \frac{1}{T-1} \sum_{j \neq i} CovSim_{i,j} \tag{3.7}$$

Since we are training models with T tasks, we take the average covariance similarity score $CovSim_i$ between task i and all other tasks. We measure $CovSim_i$ using equation 3.7 between 9 single-task models trained on individual GLUE tasks. For each task in Figure 3.7, we measure the similarity score on the MTL trained BERT$_{\text{BASE}}$ baseline, e.g., CoLA (MTL), or CA-MTL$_{\text{BERT-BASE}}$ model, e.g., MNLI (CA-MTL). Our score improvement measure is the % difference between a single task model and MTL or CA-MTL on the particular task. We find that covariance similarity increases for 9 tasks and that performance increases for 7 out 9 tasks. These measurements confirm that the Conditional Alignment is able to align task covariance, thereby helping alleviate task interference.

3.4.3 Jointly training on 8 tasks: GLUE

In Table 3.2, we evaluate the performance of CA-MTL against single task fine-tuned models, MTL as well as the other BERT-based adapters on GLUE. As in [8], MNLI$_{\text{m}}$ and MNLI$_{\text{mm}}$ are treated as separate tasks. Our results indicate that CA-MTL outperforms both the BASE adapter, PALS+Anneal Sampling [7], and the LARGE adapter, Adapters-256 [8].

Table 3.2 Adapters with layer freezing vs. ST/MT on GLUE test set. F1 for QQP/MRPC, Spearman for STS-B, accuracy on MNLI (m/mm), Matthew's for CoLA, accuracy otherwise. * Individual scores not available. ST=Single Task, MTL=Multitask, g.e.= greater or equal to. Results from [1][6], [2][7], [3][8].

Metric	BERT ST[1]	BERT MTL[2]	PALs +Anneal Sampl.[2]	*CA-MTL* MTL	BERT ST[1]	Adapters ST[3]	*CA-MTL* MTL
Model size	Base	Base	Base	Base	Large	Large	Large
Total params	9.0×	1.0×	1.13×	1.12×	9.0×	1.3×	1.12×
Params/task	100%	11.1%	12.5%	5.6%	100%	3.6%	5.6%
# tasks g.e. ST	—	2	4	**5**	—	3	3
CoLA	52.1	51.2	51.2	53.1	60.5	59.5	59.5
MNLI	84.6/83.4	84.0/83.4	84.3/83.5	85.9/85.8	86.7/85.9	84.9/85.1	85.9/85.4
MRPC (F1)	88.9	86.7	88.7	88.6	89.3	89.5	89.3
QNLI	90.5	89.3	90.0	90.5	92.7	90.7	92.6
QQP (F1)	71.2	70.8	71.5	69.2	72.1	71.8	71.4
RTE	66.4	76.6	76.0	76.4	70.1	71.5	79.0
SST-2	93.5	93.4	92.6	93.2	94.9	94.0	94.7
STS-B	85.8	83.6	85.8	85.3	86.5	86.9	87.7
Average	79.6	79.9	80.4	**80.9**	82.1	80.0	**82.8**

Against single task (ST) models, CA-MTL is 1.3% higher than BERT$_{\text{BASE}}$, with 5 out 9 tasks equal or greater performance, and 0.7% higher than BERT$_{\text{LARGE}}$, with 3 out 9 tasks equal or greater performance. ST models, however, need 9 models or close to 9× more parameters for all 9 tasks. We noted that CA-MTL$_{\text{BERT-LARGE}}$'s average score is driven by strong RTE scores. While RTE benefits from MTL, this behavior may also be a side effect of layer freezing. In Table B.4, we see that CA-MTL has gains over ST on more and more tasks as we gradually unfreeze layers.

3.4.4 Transfer to New Tasks

In Table 3.3 we examine the ability of our method to quickly adapt to new tasks. We performed domain adaptation on SciTail [38] and SNLI [37] datasets, using a CA-MTL$_{\text{BASE}}$ model trained on GLUE and a new linear decoder head. We tested several pretrained and randomly initialized task embeddings in a zero-shot setting. The complete set of experiments with all task embeddings can be found in the Appendix, Section B.1.2.

Table 3.3 Domain adaptation results on dev. sets for *BASE* models. [1][9], [2][10]

% data used	SciTail				SNLI			
	0.1%	1%	10%	100%	0.1%	1%	10%	100%
BERT$_{\text{BASE}}$[1]	51.2	82.2	90.5	94.3	52.5	78.1	86.7	91.0
MT-DNN[1]	81.9	88.3	91.1	95.7	81.9	88.3	91.1	95.7
MT-DNN$_{\text{SMART}}$[2]	82.3	88.6	91.3	**96.1**	82.7	86.0	**88.7**	**91.6**
CA-MTL$_{\text{BERT}}$	**83.2**	**88.7**	**91.4**	95.6	**82.8**	**86.2**	88.0	91.5

We then selected the best task embedding for our results in Table 3.3. STS-B and MRPC MTL-trained task embeddings performed best on SciTail and SNLI respectively. CA-MTL$_{\text{BERT-BASE}}$ has faster adaptation than MT-DNN$_{\text{SMART}}$ [10] as evidenced by higher performances in low-resource regimes (0.1% and 1% of the data). When trained on the complete dataset, CA-MTL$_{\text{BERT-BASE}}$ is on par with MT-DNN$_{\text{SMART}}$. Unlike MT-DNN$_{\text{SMART}}$ however, we do not add context from a semantic similarity model – MT-DNN$_{\text{SMART}}$ is built off HNN [155]. Nonetheless, with a larger model, CA-MTL surpasses MT-DNN$_{\text{SMART}}$ on the full SNLI and SciTail datasets in Table 3.6.

3.4.5 Jointly training on 24 tasks: GLUE/Super-GLUE, MRQA and WNUT2017

Effects of Scaling Task Count. In Figure 3.8 we continue to test if CA-MTL mitigates *task interference* by measuring GLUE average scores when progressively adding 9 GLUE tasks, 8 Super-GLUE tasks [23], 6 MRQA tasks [30].

Tasks are described in Appendix section B.1.6. The results show that adding 23 tasks drops the performance of our baseline MTL BERT$_{\text{BASE}}$ (π_{rand}). MTL BERT increases by 4.3% when adding MRQA but, with 23 tasks, the model performance drops by 1.8%. The opposite is true when CA-MTL modules are integrated into the model. CA-MTL continues to show gains with a large number of tasks and surpasses the baseline MTL model by close to 4% when trained on 23 tasks.

24-task CA-MTL. We *jointly* trained large MTL baselines and CA-MTL models on GLUE/Super-

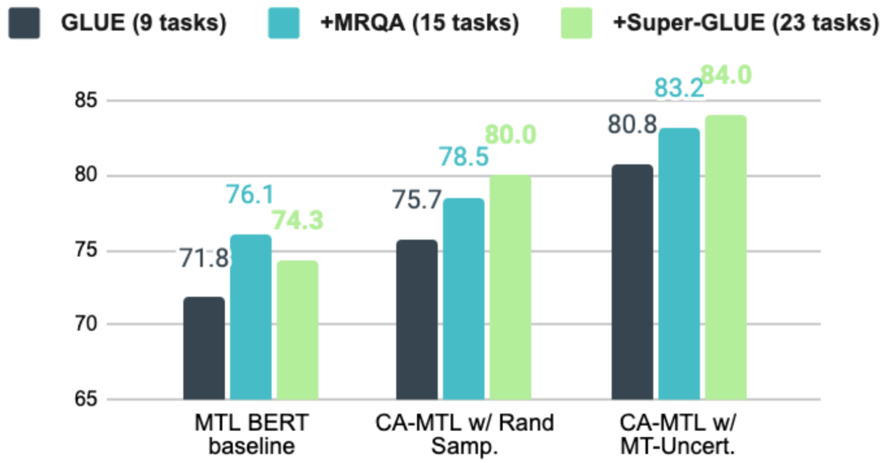


Figure 3.8 Effects of adding more datasets on avg GLUE scores. Experiments conducted on 3 epochs. When 23 tasks are trained jointly, performance of CA-MTL$_{\text{BERT-BASE}}$ continues to improve.

GLUE/MRQA and Named Entity Recognition (NER) WNUT2017 [156]. Since some dev. set scores are not provided and since RoBERTa results were reported with a median score over 5 random seeds, we ran our own single seed ST/MTL baselines (marked "ReImp") for a fair comparison. The dev. set numbers reported in [15] are displayed with our baselines in Table B.3. Results are presented in Table 3.4.

Table 3.4 24-task CA-MTL vs. ST and vs. 24-task MTL with frozen layers on GLUE, SuperGLUE, MRQA and NER development sets. ST=Single Task, MTL=Multitask, g.e.= greater or equal to. Details in section B.1.3.

Model	Task Grouping				Avg	# tasks	Total
	GLUE	SuperGLUE	MRQA	NER		e.g. ST	Params
BERT-LARGE models							
ST$_{\text{ReImp}}$	84.5	68.9	<u>79.7</u>	<u>54.1</u>	76.8	—	24×
MTL$_{\text{ReImp}}$	83.2	72.1	77.8	42.2	76.4	9/24	1×
CA-MTL	<u>86.6</u>	<u>74.1</u>	79.5	49.0	<u>79.1</u>	**17/24**	1.12×
RoBERTa-LARGE models							
ST$_{\text{ReImp}}$	88.2	76.5	<u>83.6</u>	<u>57.8</u>	81.9	—	24×
MTL$_{\text{ReImp}}$	86.0	78.6	80.7	49.3	80.7	7/24	1×
CA-MTL	<u>89.4</u>	<u>80.0</u>	82.4	55.2	<u>83.1</u>	**15/24**	1.12×

We notice in Table 3.4 that even for large models, CA-MTL provides large gains in performance on average over both ST and MTL models. For the BERT based models, CA-MTL provides 2.3% gain over ST and higher scores on 17 out 24 tasks. For RoBERTa based models, CA-MTL provides 1.2% gain over ST and higher scores on 15 out 24 tasks. We remind the reader that this is achieved with a single model. Even when trained with 16 other tasks, it is interesting to note that the MTL baseline perform better than the ST baseline on Super GLUE where most tasks have a small number of samples. Also, we used NER to test if we could still outperform the ST baseline on a token-level task, significantly different from other tasks. Unfortunately, while CA-MTL performs significantly better than the MTL baseline model, CA-MTL had not yet overfit on this particular task and could have closed the gap with the ST baselines with more training cycles.

Table 3.5 Our 24-task CA-MTL vs. other large models on GLUE. F1 is reported for QQP/MRPC, Spearman's corr. for STS-B, Matthew's corr. for CoLA and accuracy for other tasks. *Split not available. **Uses intermediate task fine-tuning + ST.

Model	**GLUE tasks**								Avg
	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	
BERT-LARGE based models on Dev set.									
MT-DNN	63.5	87.1/86.7	91.0	92.9	89.2	83.4	94.3	90.6	85.6
STILTS **	62.1	86.1*	92.3	90.5	88.5	83.4	93.2	90.8	85.9
BAM!	61.8	87.0*	–	92.5	–	82.8	93.6	89.7	–
24-task CA-MTL	63.8	86.3/86.0	92.9	93.4	88.1	84.5	94.5	90.3	**86.6**
RoBERTa-LARGE based models on Test set.									
RoBERTA** with Ensemble	67.8	91.0/90.8	91.6	95.4	74.0	87.9	97.5	92.5	**87.3**
24-task CA-MTL	62.2	89.0/88.4	92.0	94.7	72.3	86.2	96.3	89.8	85.7

Comparisons with other methods. In Table 3.5, CA-MTL$_{\text{BERT}}$ is compared to other Large BERT based methods that either use MTL + ST, such as MT-DNN [9], intermediate tasks + ST, such as STILTS [14] or MTL model distillation + ST, such as BAM! [136]. Our method scores higher than MT-DNN on 5 of 9 tasks and by 1.0 % on avg. Against STILTS, CA-MTL realizes a 0.7 % avg. score gain, surpassing scores on 6 of 9 tasks. We also show that CA-MTL$_{\text{RoBERTa}}$ is within only 1.6 % of a RoBERTa ensemble of 5 to 7 models per task and that uses intermediate tasks. Using our 24-task CA-MTL large RoBERTa-based model, we report NER F1 scores on the WNUT2017 test set in Table 3.6a. We compare our result with RoBERTa$_{\text{LARGE}}$ and XLM-R$_{\text{LARGE}}$ [157] the current state-of-the-art (SOTA). Our model outperforms XLM-R$_{\text{LARGE}}$ by 1.6%, reaching a new state-of-the-art. Using domain adaptation as described in Section 3.4.4, we report results on the SciTail test set in Table 3.6b and SNLI test set in Table 3.6b. For SciTail, our model matches the current SOTA[4] ALUM [158], a RoBERTa large based model that additionally uses the SMART [10] fine-tuning

[4] https://leaderboard.allenai.org/scitail/submissions/public on 09/27/2020

method. For SNLI, our model outperforms SemBert, the current SOTA[5].

For both SciTail and SNLI, we found interesting to compare against MT-DNN. Although MT-DNN is based on a BERT large model, it reaches competitive results on both tasks and it's an MTL based model that uses similar adaptation technique. Our model performs better than MT-DNN. However, it isn't clear if this increase of performance is only attributed to the RoBERTa model.

Note that [157] used $\text{RoBERTa}_{\text{LARGE}}$ [15] and $\text{XLM-R}_{\text{LARGE}}$ [159] as large model baselines. $\text{CA-MTL}_{\text{RoBERTa-LARGE}}$ outperforms $\text{XLM-R}_{\text{LARGE}}$ by 1.6% WNUT2017 F1 score.

Table 3.6 CA-MTL test performance vs. SOTA.

(a) **WNUT2017**	F1
$\text{RoBERTa}_{\text{LARGE}}$	56.9
$\text{XLM-R}_{\text{LARGE}}$	57.1
$\text{CA-MTL}_{\text{RoBERTa}}$	**58.0**

(b) **SciTail**	% Acc
MT-DNN	94.1
$\text{ALUM}_{\text{RoBERTa}}$	96.3
$\text{ALUM}_{\text{RoBERTa-SMART}}$	**96.8**
$\text{CA-MTL}_{\text{RoBERTa}}$	**96.8**

(c) **SNLI**	% Acc
MT-DNN	91.6
$\text{MT-DNN}_{\text{SMART}}$	91.7
SemBERT	91.9
$\text{CA-MTL}_{\text{RoBERTa}}$	**92.1**

3.5 Conclusion

We believe that our experiments here have helped demonstrate the potential of task conditioned adaptive learning within a single model that performs multiple tasks. In a large-scale 24-task NLP experiment, CA-MTL outperforms fully tuned single task models by 2.3% for BERT Large and by 1.2% for RoBERTa Large using 1.12 times the number of parameters, while single task fine-tuning approach requires 24 separately tuned single task models or 24 times the number of parameters. When a BERT vanilla MTL model sees its performance drop as the number of tasks increases, CA-MTL scores continue to climb. Performance gains are not driven by a single task as it is often the case in MTL. Each CA-MTL module that adapts a Transformer model is able to reduce performance variances between tasks, increasing average scores and aligning task covariances. This evidence shows that CA-MTL is able to mitigate task interference and promote more efficient parameter sharing. We showed that MT-Uncertainty is able to avoid degrading performances of low resource tasks. Tasks are sampled whenever the model sees entropy increase, helping avoid catastrophic forgetting. Overall, CA-MTL offers a promising avenue to dynamically adapt and modularize knowledge embedded in large monolithic pretrained models. Extending such ideas will be an objective

[5] https://nlp.stanford.edu/projects/snli/ on 09/27/2020

for future work.

CHAPTER 4 ARTICLE 3: INTERACTIVE-CHAIN-PROMPTING: AMBIGUITY RESOLUTION FOR CROSSLINGUAL CONDITIONAL GENERATION WITH INTERACTION

Jonathan Pilault[1,2,3*] Xavier Garcia[1] Arthur Bražinskas[4] Orhan Firat[1]

Accepted in The 13th International Joint Conference on Natural Language Processing and The 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL), Publication Date: November 2023

[1]Google DeepMind [2]Mila [3]Polytechnique Montreal [4]Google Research, XGen Team

pilaultj@mila.quebec {xgarcia, orhanf}@deepmind.com

*First Author[1]

Abstract

Crosslingual conditional generation (e.g., machine translation) has long enjoyed the benefits of scaling. Nonetheless, there are still issues that scale alone may not overcome. A source query in one language, for instance, may yield several translation options in another language without any extra context. Only one translation could be acceptable however, depending on the translator's preferences and goals. Choosing the incorrect option might significantly affect translation usefulness and quality. We propose a novel method *interactive-chain prompting* — a series of question, answering and generation intermediate steps between a *Translator* model and a *User* model — that reduces translations into a list of subproblems addressing ambiguities and then resolving such subproblems before producing the final text to be translated. To check ambiguity resolution capabilities and evaluate translation quality, we create a dataset exhibiting different linguistic phenomena which leads to ambiguities at inference for four languages. To encourage further exploration in this direction, we **release** all datasets. We note that *interactive-chain prompting*, using eight interactions as exemplars, consistently surpasses prompt-based methods with direct access to background information to resolve ambiguities.

[1] Jonathan Pilault a contribué dans la conception des algorithmes, l'experimentation, l'analyse des résultats, amélioration de l'efficacité du logiciel, analyse théorique, rédaction de l'article, recherche bibliographique et la réponse aux réviseurs de conférence.

4.1 Introduction

Transformer Language Models (LM, [160]) pretrained on large corpora have achieved outstanding results in a variety of NLP benchmarks [161, 162]. Scaling the number of parameters, the size of the pretraining dataset, and the amount of computing budget gives Language Models better sample efficiency and ability to generalize for many tasks [163, 162, 164, 165, 166, 167]. However, for tasks such as commonsense and symbolic reasoning, where the solution requires multistep computation, or crosslingual conditional generation such as Neural Machine Translation (NMT), where there could be more than one plausible prediction for a given source sequence, scale alone may not be sufficient to achieve high accuracy [168, 169].

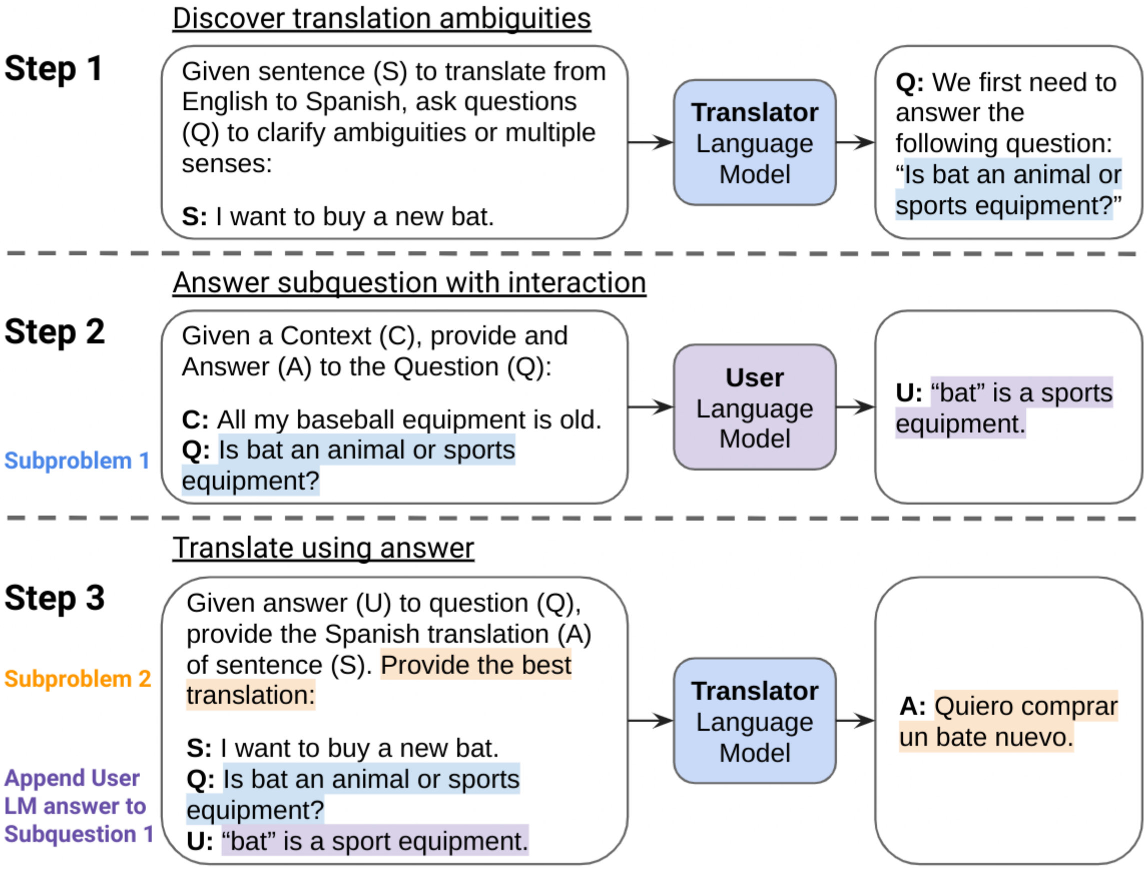

Discover translation ambiguities
Step 1
Given sentence (S) to translate from English to Spanish, ask questions (Q) to clarify ambiguities or multiple senses:
S: I want to buy a new bat.
Translator Language Model
Q: We first need to answer the following question: "Is bat an animal or sports equipment?"
Answer subquestion with interaction
Step 2
Subproblem 1
Given a Context (C), provide and Answer (A) to the Question (Q):
C: All my baseball equipment is old.
Q: Is bat an animal or sports equipment?
User Language Model
U: "bat" is a sports equipment.
Translate using answer
Step 3
Subproblem 2
Append User LM answer to Subquestion 1
Given answer (U) to question (Q), provide the Spanish translation (A) of sentence (S). Provide the best translation:
S: I want to buy a new bat.
Q: Is bat an animal or sports equipment?
U: "bat" is a sport equipment.
Translator Language Model
A: Quiero comprar un bate nuevo.

Figure 4.1 Interactive-Chain-Prompting (*InterCPt*).

Chain-of-thought [170] and least-to-most [171] methods have demonstrated, by prompting a (large-)LM such as PaLM [172], that breaking down a task into subproblems that are solved sequentially greatly improves the quality of the final prediction. Such methods demonstrate that producing intermediate sub-results that address specific aspects of a bigger problem significantly improves performance on tasks like arithmetic, math word problems, and symbolic manipulation.While studies have investigated the translation capabilities of PaLM with various prompting strategies [173, 174], prompting large and general purpose LMs such as PaLM to identify and solve subproblems in crosslingual conditional generation tasks such as NMT has not yet been fully explored.

Our approach, *Interactive-Chain-Prompting* (*InterCPt*), sequentially solves translation subproblems before generating a final translation prediction. As shown in Figure 4.1, we first detect ambiguities in translation queries, then we resolve these ambiguities via question-answer interactions, and finally we generate translations. *InterCPt* departs from other prompt-based techniques that sequentially solve subproblems in two fundamental ways: (1) the subproblems are related but considerably different to the main task and (2) the solutions to subproblems requires interaction with another LLM. In this paper, we will look at how intermediate computation steps and interaction might overcome a typical problem in automated systems when a user's ambiguous query leads to a large number of viable and potentially inaccurate answers. In translation, for example, selecting the incorrect prediction has a significant impact on translation quality as illustrated in Fig. 4.2.

InterCPt has several advantages. First, the LM is able to identify and ask questions about translation query ambiguities with only a few in-context exemplars and no finetuning. This is crucial since large corpora with specific target ambiguities, labels to classify each ambiguity subtypes (i.e. feminine/masculine for gender or formal/informal for formality) and context are not common and are typically low-resource. Then, without readily available context, we rely on the *User* to disambiguate translation queries. In the absence of additional background information or context, there are limited options to solve ambiguities. Interaction with the *User* stands as a logical way to collect clarifying information. This interaction also benefits from multiple computation steps where ambiguity resolution leads to a more precise final prediction. By resolving a few high-leverage ambiguities up front, we reduce retries and long prompts, improving both quality and wall-clock efficiency. Finally, the question-answer-translation interaction improves transparency and makes it easier to debug translation systems since we can assess the reasoning chain that led to an error [175]. For NMT, there are two main questions to consider to make the most of out of intermediate computation steps:

A) What subproblem are we trying to solve? Multistep reasoning tasks can often be explicitly decomposed into subproblems: ambiguity detection, disambiguation via Q&A and translation. For NMT, decomposing the translation task is not trivial. We assume in this work that our subproblems are ambiguities which arise when translating. As seen in Fig. 4.1, the first step in *InterCPt* is to discover and resolve the translation ambiguity subproblem. We study five types of ambiguities: polysemous words, pronoun resolution, formality, gender-neutral names and neutral professions. Since datasets that cover multiple translation ambiguities and language pairs while providing context are rare, we create our own datasets (see Table C.1 in Section C.1.4 for an overview of other publicly available datasets).

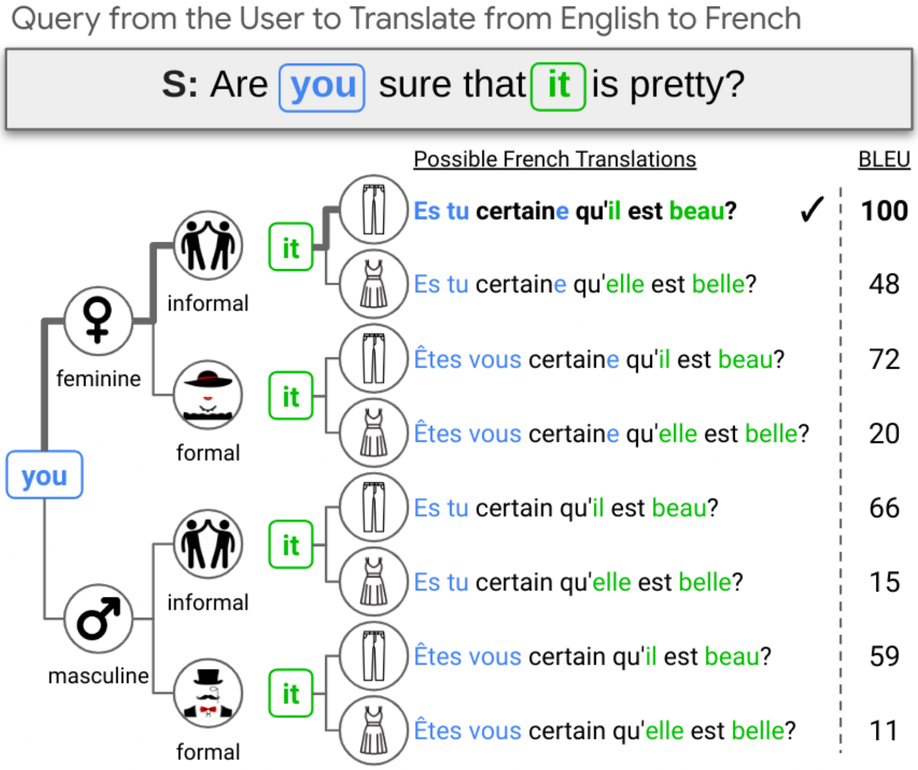

Query from the User to Translate from English to French
S: Are you sure that it is pretty?
Possible French Translations
BLEU
you
feminine
masculine
informal
formal
informal
formal
it
it
it
it
Es tu certaine qu'il est beau? ✓
100
Es tu certaine qu'elle est belle?
48
Êtes vous certaine qu'il est beau?
72
Êtes vous certaine qu'elle est belle?
20
Es tu certain qu'il est beau?
66
Es tu certain qu'elle est belle?
15
Êtes vous certain qu'il est beau?
59
Êtes vous certain qu'elle est belle?
11

Figure 4.2 Translation queries with multiple possible predictions. Correctly solving subproblems around ambiguities with **you** and **it** greatly affects the *BLEU* [40] translation metric.

B) Where do answers to subquestions come from? When we apply least-to-most prompting to math word problems for example, the answers to subquestions can often be derived from the problem's text. It is not necessarily the case for NMT where the query may not contain enough context to resolve ambiguities. As seen in Fig. 4.2, English sentence 'S' does not contain enough information about "you" and "it". The incorrect prediction made by a model leads to large variations in translation quality scores. With more context, the model may have the necessary information to narrow down possible predictions. However, in industrial applications, translation queries are often too short [176] or additional context is not existent. In this work, we automate interaction between a *PaLM Translator* model, that detects ambiguities, asks clarifying questions and translates, and a *PaLM User* model, that has access to context and answers questions. Both models engage in a multiturn dialog to zero-in on a narrower set of predictions. We argue that a type of question-answer interaction with a "user" is necessary to resolve ambiguous queries, especially when a user (1) is unfamiliar with the main task and may not possess the skills to choose from many model prediction options; (2) knows how to answer simple pointed questions about a query but may not be able or willing to decide and add appropriate context on the fly.

This work marks Large-LM's potential to learn, with a few in-context examples, how to use natural language answers to deliver results closer to a user's intent. Our contributions are the following:

1. We propose *InterCPt*, a new way to design crosslingual conditional generation systems that disambiguate queries via interaction.
2. We release *AmbigMT*, a new dataset with five specific types of ambiguities covering four languages.
3. We show that *InterCPt* achieves better translation performance and ambiguity resolution (Section 4.4) and improved generalization on zero-shot ambiguities (Section 4.5) over strong baselines.
4. We provide analysis on interactions and evidence that *InterCPt* abilities emerge with scale (Section 4.5).

4.2 Interactive-Chain-Prompting (InterCPt)

When interacting with a model, a user may have some well-conceived query in mind that is inadvertently under-specified. For example, a monolingual English speaker may be unaware that the pronoun "you" in a sentence can lead to formal or informal constructs in other languages and may therefore not provide additional information on the level of formality needed to adequately translate the text.

A human translator, when asked to translate queries with "you", may want to first probe the user's latent context about the query by asking clarifying questions. In doing so, the human translator can use the answers to better align the translation to a User's request and context. Our method endows language models (LMs) with the ability to generate a similar chain of interactions between a Translator LM and a User LM as seen in Fig. 4.1. In real applications, it is expected that a human replaces the User LM. *InterCPt* uses in-context exemplars to resolve ambiguities before completing the crosslingual conditional generation task that the model is originally asked to do.

The three step reasoning chain (see Fig. 4.1):

1. **The first step is for identifying ambiguities.** The prompt in this step always contains the same constant exemplars, showing multiple queries to translate and questions about each query's ambiguities. During inference, the *Translator* LM uses the prompt to generate a pointed question that identifies the specific ambiguity.
2. **The second step is for resolving ambiguities.** The prompt in this step contains exemplars answering the question to the ambiguity subproblems in step one. The *User* LM answers each question using additional information from the provided context. In real life applications, we assume that a real user has similar background information about the text to be translated.
3. **The third step is for translating.** Generated questions and answers are appended to the prompt in step 1 before the final translation is produced. Constant prompts in this step demonstrate how to translate in the specified target language using only details provided by the *User* LM and no-context. During inference, the *Translator* LM uses the prompt to generate the translation.

Dataset	***en* Query**	**Context**	***x* Target**	Δ B
"it" resolution	He has read **it** to me so many times that I've learnt **it** by heart.	– I remember when the **postcard** came, Ernesto was so pleased. – He said: "Look what my Rosetta has written to me".	Me **la** sé de memoria de tanto **leerla**.	-44
Polysemy	**head**	If you don't feel well, **head** home.	**先**	-100
Formality	The closer **you** can get to him, the better.	– I'm aware of the risks, **Master Jedi**, but I know **you** can regain Clovis' trust.	Plus **vous serez** proche de lui, mieux **cela** sera.	-58
Gender neutral names	**Blair** should be wrapping up [pr] breakfast with Beatrice.	– I have **her** doorman on retainer. – There's a fine line between surveillance and stalking.	Blair sollte **ihr** Frühstück mit Beatrice haben.	-40
Neutral professions	[pr] worked previously as a businesswoman, accountant, and bank executive.	**Margaret** Mhango Mwanakatwe is a Zambian politician [...]. **She** was the director for business development [...]	Previamente, trabajó como **empresaria**, **contadora** y **ejecutiva** bancaria.	-70

Table 4.1 *AmbigMT* examples for each ambiguity for target language *x*. Δ B is the *BLEU* performance drop from 100 if the highlighted ambiguity is not resolved.

In this section, we introduce *AmbigMT*, a dataset that covers four language pairs, for translations from English into French (en-fr), German (en-de), Spanish (en-es) or Japanese (en-ja)

— 18 sub-tasks in total. The code and datasets are released **here**. The parallel translation corpora contain five types of ambiguities: "it" resolution, formality, polysemy, gender[2] neutral names, neutral professions. Unless otherwise specified, all datasets include 1000 diverse samples for each {en-fr, en-de, en-es, en-ja} language pair extracted from Opensubtitles corpora [177]. In Section C.1.4 of the Appendix, we provide more details on datasets and describe the heuristics to identify ambiguities in each language.

"it" resolution data contains English sentences where the pronoun "it" does not clearly refer to a noun within the query. In English, the pronoun "it" is a singular, neuter and impersonal pronoun. In other languages, "it" may translate into gender specific pronouns (either feminine or masculine) or get dropped entirely from the sentence. The choice depends on what the pronoun refers to. To correctly translate, the model must first determine what "it" is. In the first example of table 4.1 where the target language x is Spanish, knowing that "it" is a postcard, or *una tarjeta postal* in Spanish, disambiguates gender in the translation. While the gender affects two words in the target sentence, the wrong gender choice is not only qualitatively inappropriate but also decreases quality metrics (44 *BLEU* score drop from 100).

Polysemy is a dataset that contains words that have multiple meanings and the query is insufficiently informative to zero-in on a specific sense. The context uses the word within a sentence to provide the necessary background information. In the second example of Table 4.1 where the target language x is Japanese, the context shows that "head" is a verb. In conjunction with the noun "home", we disambiguate "head" as "to move in the direction of". In the absence of such context, "head" has various senses such as "upper part of the body", "side of a coin", "end of a hammer or tool", "a toilet on a boat", "to hit the ball with the head", "to lead".

Formality is a dataset where English queries contain the pronoun "you". In the target languages studied, "you" can be formal or informal. As seen in the third example of table 4.1 where the target language x is French, the speaker addresses the listener "you" as "Master Jedi" in the context, a title implying a formal style of politeness. The formality is ambiguous without the context and may impact the generated translation quality. Indeed, an incorrect choice in formality level changes "vous serez" to "tu seras" and "cela" to "ça", decreasing *BLEU* scores by 58 points from 100.

[2] Please note that due to the lack of large translation corpora with various genders and the complexity in creating non-binary gender datasets, our data is limited to feminine and masculine.

Gender Neutral Names data includes queries where the name is gender neutral and ambiguous. The fourth example in table 4.1 shows a query where the name “Blair” is gender neutral. In this dataset, we replace gendered pronouns in the English query by the token *[pr]* to remove hints about gender type. From the context, the speaker employs “her” and we can infer that a feminine pronoun “ihr” should be used in the translated German text.

Neutral Professions has 600 unique samples for two language pairs. This dataset is derived from the Translated Wikipedia Biographies dataset[3] that covers {en-de, en-es}. In this dataset, the gender of typically gender-neutral professional designations is not clear from the English query alone. In the fifth example of table 4.1, the context provides additional hints that the query is talking about “Margeret”, also designated by the feminine pronoun “she”. Resolving gender allows the model to correctly translate the list of professions in the query and potentially limiting the 70 points drop in *BLEU* scores from 100.

4.3 Related Works

Prompting for Cross-Lingual Generation using Large LMs is a technique that has garnered increasing attention of late. Works on GPT-3 [160] and PaLM [172] show competitive n-shot *BLEU* translation results on WMT. The prompt demonstrations are populated with n random sentence pairs taken from the WMT training corpora and evaluated on the test corpora at inference. Orthogonal to our work, POMP [173] improves upon this PaLM-based prompting technique by explicitly optimizing for the selection of n demonstration sentence pairs and obtaining results competitive with the state-of-the-art. We used observations from [174] that demonstrate that pseudo-parallel prompt examples can improve translation quality, and that additional gains are possible by transferring knowledge from prompt examples selected under different settings. More recent work [178] using mT5 [179] investigated adding prompt-based natural language specifications to influence translated text properties such as formality level or dialect type. Experiments show that prepending textual artifacts such as “your majesty” to the English query conditions mT5 to generate translations in a formal tone. Our work prompts PaLM with n random translation pair exemplars as well. Different from previous research, we prompt with exemplars to interactively discover background knowledge or clarify ambiguities before translating.

Resolving ambiguities by asking for clarifications has been a recent topic of research, for QA and conversational search systems [180, 181, 182, 183, 184, 185]. Departing from such

[3] https://ai.googleblog.com/2021/06/a-dataset-for-studying-gender-bias-in.html

methods, *InterCPt* does not produce sentences from a preset list of questions but is generated from a large LM without constrain. Concurrently to our work, [186] explored finetuning GPT-3 to generate clarifying questions and provide answers using human generated data from AmbigQA [187] for open-domain QA. Another GPT-3 model simulates the user and generates answers while conditioned on ground-truth clarification questions. In contrast, our prompt-based method only needs few-shot demonstrations. Further, our simulated user does not rely on ground-truth clarification questions to provide an answer, which could be more realistic for a number of applications (including QA, text simplication, code generation).

Interactive Machine Learning [188, 189, 190] is an approach where information is interactively and iteratively supplied to a learning system. In prior interactive translation work, machine interactivity has assisted translators in writing translations by displaying automated word suggestions that update incrementally [191, 192]. The approach however is limited by drop-down menu options and requires a certain level of sophistication from the user in the *target language*. Our approach discovers preferences and background knowledge about an input query in the *source language* and more flexibly adapts translations according to a user's natural language response. The interaction is similar to Conversational AI systems where user utterances influence generated outputs. Task or goal oriented conversational AI systems [193, 194, 195] are typically deployed to answer knowledge-based questions, seek information or solve basic queries (e.g. making reservations, purchase an item). To our knowledge, our work is the first to explore conversational interaction in cross-lingual generation.

4.4 Experimental Setup and Results

In this section, we present the main cross-lingual generation results of *InterCPt* for formality, "it" resolution and polysemy ambiguity resolution subtasks.

Setup. We use PaLM [172], a 540B-parameter decoder-only LM pretrained on primarily English-centric data with ∼20% of the data obtained from non-parallel multilingual corpora. The *generalist* prompt template is composed of two formality, three polysemy and three "it" resolution exemplars. All prompt-based methods are 8-shot with the same source sentences S to translate and corresponding translated sentences A in the target language. Each target language has its own prompt template since A differs with every language. The simulated LM user is based on a single English-only 8-shot prompt template for all target languages. Example 4.4.1 shows the structure of an LM user prompt exemplars for polysemy. A complete

Lang. Pairs	Method	Formality			“it” resolution			Polysemy			
		bleu	**bleurt**	**F-Acc.**	**bleu**	**bleurt**	**G-Acc.**	**Hit@3**	**Hit@10**	**B@3**	**B@10**
en→es	*InterCPt*	**36.3**†	**77.9**†	**67%**	**33.6**†	**78.9**†	**77%**	**46%**	**48%**	**54.6**†	**56.8**†
	LLMwCxt	34.7	77.1	64%	30.8	77.2	68%	40%	46%	46.9	55.1
	LLMnoExtra	34.6	77.0	62%	29.6	75.9	63%	33%	40%	44.9	51.0
	GTranslate	31.4	75.3	50%	27.5	73.0	54%	—	—	—	—
en→fr	*InterCPt*	**39.1**†	**70.6**	**72%**	**35.3**†	**71.7**†	**73%**	**46%**	**48%**	**46.9**†	**48.5**†
	LLMwCxt	36.4	69.9	65%	33.5	68.4	68%	36%	40%	40.1	44.7
	LLMnoExtra	35.7	69.2	63%	32.3	66.7	66%	33%	37%	38.1	41.8
	GTranslate	30.7	67.4	58%	29.1	65.4	61%	—	—	—	—
en→de	*InterCPt*	**35.8**†	**75.0**	**69%**	**24.0**†	**76.0**	**75%**	**43%**	**45%**	**45.1**†	**47.6**†
	LLMwCxt	33.6	74.6	61%	22.4	75.0	69%	35%	39%	36.1	44.9
	LLMnoExtra	32.5	74.4	62%	22.8	73.2	63%	32%	35%	36.7	41.3
	GTranslate	27.5	72.3	53%	22.1	73.0	59%	—	—	—	—
en→ja	*InterCPt*	**28.6**†	**69.7**†	**67%**	**23.1**†	**72.4**†	**74%**	**41%**	**44%**	**44.7**†	**47.0**†
	LLMwCxt	26.3	68.0	60%	21.4	70.8	67%	34%	38%	35.8	43.8
	LLMnoExtra	25.9	67.4	61%	21.2	70.3	61%	30%	33%	34.6	37.0
	GTranslate	23.5	66.7	50%	19.9	68.6	52%	—	—	—	—

Table 4.2 Translation results using an 8-shot generalist template that contains exemplars for formality, “it” resolution and polysemy ambiguity types. F-Acc = formality accuracy, G-Acc = gender accuracy, B@n = *bleurt*@n. *bleu* and *bleurt* results for *InterCPt* labelled with † are significantly better than all other systems based on pair-wise significance testing [11] with p = 0.05.

overview of all prompts and exemplars used in experiments can be found in Sections C.1.8 for the User LM and Sections C.1.9 for the generalist Translator LM.

Example 4.4.1. *Given a Context (C), provide an Answer (A) to the Question (Q):*

S: *about*

C: *About 2% of the households are enumerated using the canvasser method.*

Q: *Is “about” an adverb that means approximately, near or a preposition that means regarding, over, surrounding?*

A: *“about” means approximately.*

Baselines. Our main baselines were chosen to compare the cross-lingual generation abilities of large multipurpose LMs given interaction, context or no additional information. Please note that, to the best of our knowledge, there are no other baselines that (1) explore large multipurpose LM’s capability on contextualized (or interactive) multilingual translation; (2) do not require finetuning on large datasets.

LLMwCxt is the only PaLM-based prompt method that benefits from having *all of the background information required* to resolve ambiguities. Since this baseline has access to all information and the same in-context translation examples, it is strongest possible baseline to compare against for ambiguity resolution. *LLMwCxt* has a prompt with exemplars formulated as the one in example 4.4.2. In the example, references to **you** and **it** are directly accessible in context *C*.

LLMnoExtra is a PaLM-based prompt method that does *not* receive additional information to resolve ambiguities. This baseline is not only of interest for performance comparison and to evaluate model bias but also it can provide insights on the usefulness of additional background information to disambiguate queries. The structure of a *LLMnoExtra* exemplar is similar to example 4.4.2 without the context *C*. The model must translate the source sentence S in the target language without knowing details about "i" or the level of formality to employ for "you".

GTranslate is a commercially available multilingual and multipurpose baseline queried using the Google Cloud Translation v2 model[4]. This baseline allows us to set performance expectations that *LLMnoExtra* model should reach.

Example 4.4.2. *Given context (C), Translate (S) from English to French:*
S: *Are* ***you*** *sure that* ***it*** *is pretty?*
C: *She was trying on a new* ***hat****. Looking at herself in the mirror, she asked her* ***friend Isabelle****.*
A: ***Es-tu*** *certaine qu'****il*** *est* ***beau****?*

Metrics. Our evaluation includes the standard *BLEU* and *BLEURT* [196] automatic translation quality metrics as well as additional measures that assess specific ambiguity resolution capabilities. For formality, we use a rule-based classifier to quantify generated sentence formality levels (F-Acc) in the target language. We discuss details of the heuristics in Appendix C.1.13. Note that the formality classifier is based on the formality data creation scripts that allowed us to automatically identify formal and informal sentences in the source corpus. For "it resolution", we found that the PaLM 62B-parameter model was surprisingly accurate at identifying translated sentence genders (G-Acc). As seen in Table C.3 of Appendix C.1.13, PaLM 62B achieves 97% and 93% accuracy in classifying samples of generated translations for Spanish and French respectively. For polysemy, we found that exact match metrics did not fully describe the performance of models. Whenever the model generated a synonym of the ground truth, the exact match metric would not consider the prediction correct. The *LLMnoExtra* polysemy exemplars are a comma-separated list of synonyms. Our hit@n measures whether the ground truth exists in the first n generated words. For

[4] https://translate.google.ca/

example, if the model outputs the list of Spanish words [“aproximadamente”, “cerca de”, “alrededor de”, “casi”, “más o menos”], for $n = 3$, hit@3 would return a match for a ground truth target “cerca de” and no-match for a ground truth target “casi”. To supplement the hit@n metric, we also report results of a new metric that we call *BLEURT*@n (B@n) which returns the highest *BLEURT* score of the first n generated word phrases. Since *BLEURT* captures the non-trivial semantic similarities between words using its contextual representations from BERT, we found that the metric better measures if correct synonyms were generated by the model. Note that we did not report the *GTranslate* hit@n or B@n numbers since the API only provides single word outputs.

Discussion. Our test results for en-es, en-fr, en-de and en-ja are summarized in Table 4.2. We first notice that *InterCPt* surpasses all other baselines. Surprisingly, *LLMwCxt*, even with all the necessary background to resolve ambiguities, significantly lags behind *InterCPt* on F-Acc. for formality, G-Acc. for “it resolution” and both hit@3 and B@3 for polysemy. This results suggests that the multistep computation approach of fist resolving the ambiguity subproblems and then generating text has an advantage over other baselines. *BLEU* scores are also 2-3 points higher while *BLEURT* scores are only slightly higher. This suggest that *InterCPt* generates sentences syntactically much closer to the ground truth while conserving the correct semantics.

4.5 Analysis

In this section, we analyse interesting behaviors about our approach such as ambiguity generalization in Subsection 4.5.1, the importance of ambiguity resolution specialization in Subsection 4.5.2, the effects of scale for both the Translator LM in Subsection 4.5.3 and User LM in Subsection 4.5.4, an error analysis in Subsection 4.5.6 and bias in generated outputs in Subsection 4.5.5.

4.5.1 How does interaction generalize?

In Table 4.3, we provide translation test results on two held-out datasets that are described: (1) Gender Neutral Names and (2) Neutral Professions.

We use the same *generalist* prompt template as in Section 4.4 with exemplars that cover only

Pair	Method	**bleu**	**bleurt**	**G-Acc.**
Gender Neutral Names — unseen ambiguities				
en→es	*InterCPt*	**31.8**†	**74.1**†	**76%**
	LLMwCxt	29.9	72.4	66%
	LLMnoExtra	30.9	71.6	59%
	GTranslate	27.8	66.1	56%
en→fr	*InterCPt*	**31.0**	**63.5**†	**71%**
	LLMwCxt	29.5	62.6	64%
	LLMnoExtra	30.0	60.9	63%
	GTranslate	24.5	57.7	56%
en→de	*InterCPt*	**17.9**†	**72.2**	**73%**
	LLMwCxt	15.6	71.5	67%
	LLMnoExtra	15.2	70.8	61%
	GTranslate	17.1	67.1	55%
en→ja	*InterCPt*	**16.1**†	**70.3**†	**71%**
	LLMwCxt	14.7	69.1	65%
	LLMnoExtra	14.4	68.3	60%
	GTranslate	14.1	66.0	54%
Neutral Professions — unseen ambiguities + unseen domain				
en→es	*InterCPt*	**37.3**	75.8	**70%**
	LLMwCxt	37.1	**76.1**	69%
	LLMnoExtra	35.5	75.7	59%
	GTranslate	37.0	72.7	56%
en→de	*InterCPt*	**14.3**	70.0	**68%**
	LLMwCxt	14.0	**71.9**	66%
	LLMnoExtra	12.2	70.0	62%
	GTranslate	13.8	67.2	54%

Table 4.3 Translation results on unseen ambiguity subproblems using the Gender Neutral Names data and with added unseen domain using the Neutral Professions data. *InterCPt* results labelled with † are significantly better with p = 0.05.

formality, "it" resolution and polysemy. Specifically, our exemplars for both the Translator LM and the User LM do not contain exemplars to resolve the gender for a person's name or profession. We observe that on the Gender Neutral Names dataset *InterCPt* performs best on *bleu* and *bleurt* and is much more able to resolve ambiguities with 6 to 10 points G-Acc improvements over *LLMwCxt*. On the Neutral Professions data, where test samples are taken from a different domain (Wikipedia biographies instead movie scripts), *LLMwCxt* and *InterCPt* have similar performances. It is possible that *LLMwCxt* benefits from additional sentences in the context to better determine the style of the output. Nonetheless, *InterCPt* provides a 1-2 point increase on G-Acc.

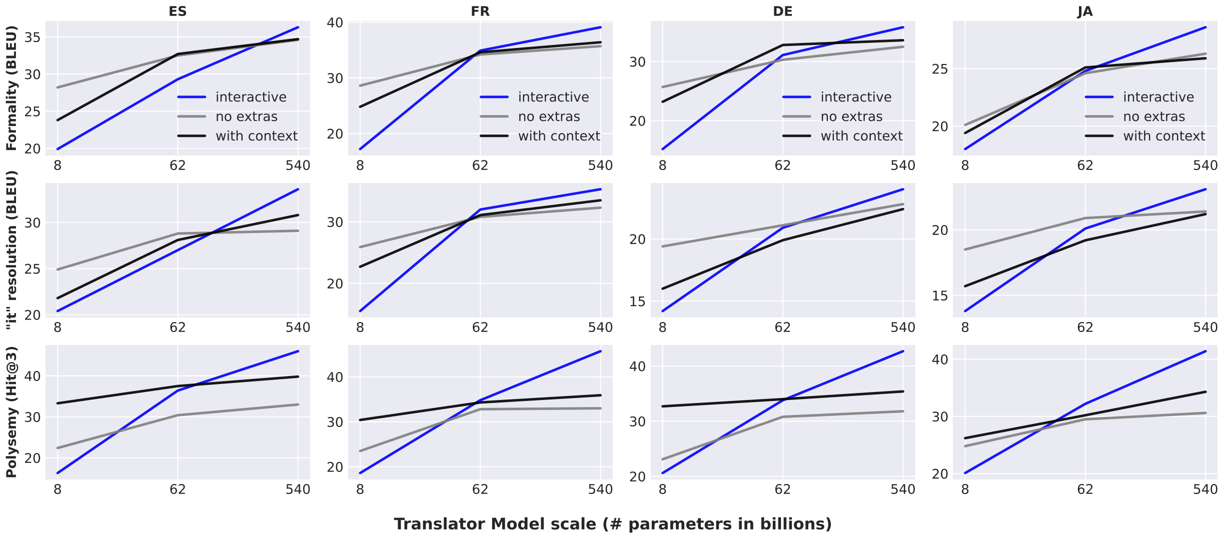


Figure 4.3 *InterCPt* enables large LMs to solve ambiguity subproblems in cross-lingual generation. The multistep disambiguate-translate capability is an emergent ability that is reached at higher parameter scales (interactive = *InterCPt*).

4.5.2 Are specialist better than generalist?

So far, we have studied a *generalist* 8-shot template covering three different types of ambiguities with at most three exemplars per ambiguity. In Fig. 4.4, we present results of *specialist* template that only covers one type ambiguity at the time (either all formality or all polysemy).

Interestingly, specialization does not seem to provide much additional benefit in resolving ambiguities as evidenced by F-Acc, Hit@3 and B@3 results that are on par and often lower than the *generalist* approach. However, the *specialist* template does have a higher BLEU score, implying greater syntactic alignment with the target translation when more ambiguity-specific exemplars are added.

4.5.3 Are interactive generation abilities emergent at scale?

We show in Fig. 4.3 for each prompt template the effects of scaling PaLM parameters on the performance of formality, "it" resolution and polysemy for Spanish (ES), French (FR), German (DE) and Japanese (JA) target languages. Please note that while we vary the parameter count (8B, 62B and 540B) of the Translator LM, the User LM is a 540B parameters

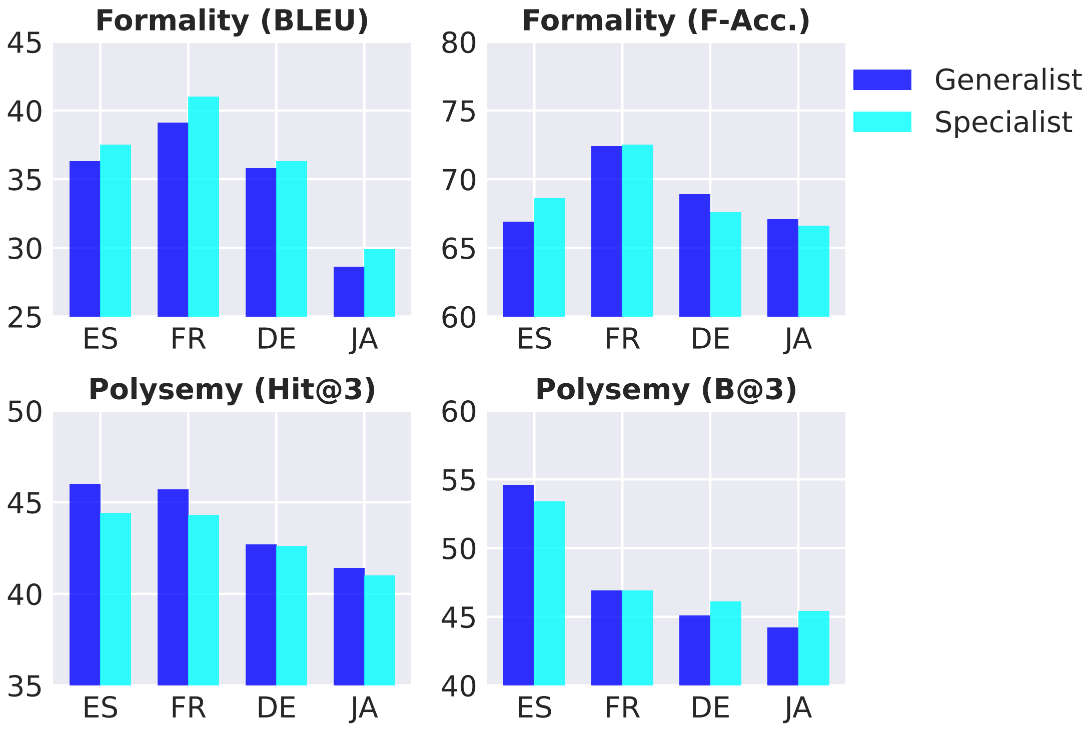


Figure 4.4 Generalist vs Specialist prompt templates for Spanish (ES), French (FR), German (DE) and Japanese (JA) targets.

PaLM model for all experiments. The plots provide interesting insights.

First, at the 8B parameter scale, *LLMNoExtra* performs best across all languages for Formality and "it" resolution across all language pairs. Neither context or interaction seem to provide benefits to translation. Second, at the 62B parameter scale, the *LLMwCxt* and *InterCPt* methods have on par performances. Context or interaction in this case are only clearly beneficial for polysemy. Third, the PaLM 540B parameter *InterCPt* outpaces other prompt-based methods across language pairs and ambiguity subproblems. At this stage, baselines scaling trend decelerates, with *scaling curves flattening*, compared to *InterCPt*. It shows that *InterCPt* is an emergent ability of model scale [167]. We conjecture that the emergent behavior of *InterCPt* is due to a better ability to ask questions and incorporate answers before generating final prediction.

4.5.4 How important is User LM scale?

While the User LM allows us to automate the evaluation of interactivity for cross-lingual generation, it is not clear if the quality of the answer to the Translator LM questions impact performance. We hypothesize that a larger User LM model would provide higher quality answers and allow the Translator LM to better generate translated text.

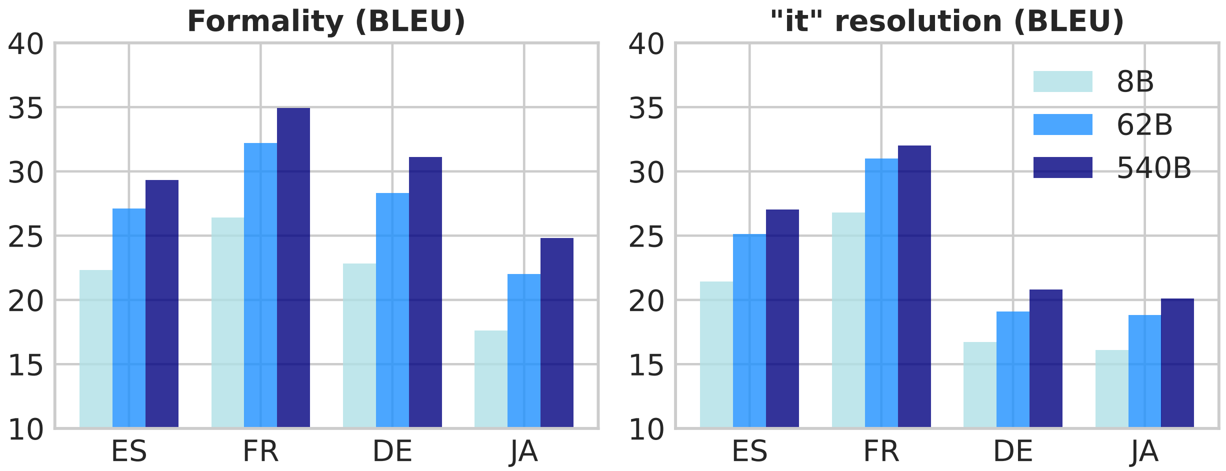


Figure 4.5 Scaling Simulated User LM improves the performance of a 62B Translator LM model.

Fig. 4.5 shows that, when the Translator LM is a 62B PaLM model, a higher parameter User LM improve overall performance. It is therefore possible that answer quality has a significant impact on translation quality and that human-generated answers can further improve overall performance.

4.5.5 Can interaction help solve bias issues?

Gender bias is a common phenomenon in automated NMT systems [197, 66, 198]. Even when there are explicit gender pronouns in the input query or in the context, NMT systems generated text tends to be masculine when translated into languages with grammatical gender [66, 198, 199, 200].

To measure gender bias, all generated translations are passed through the gender classifier for the "it" resolution balanced dataset. Similarly, to measure formality bias, generated translations are passed through the formality classifier for the formality balanced dataset. NMT systems can also suffer from formality bias [201]. However, we notice that *InterCPt* is much closer to evenly producing masculine and feminine sentences. Our results shows that interactive ambiguity resolution via multistep computation better addresses gender and formality biases.

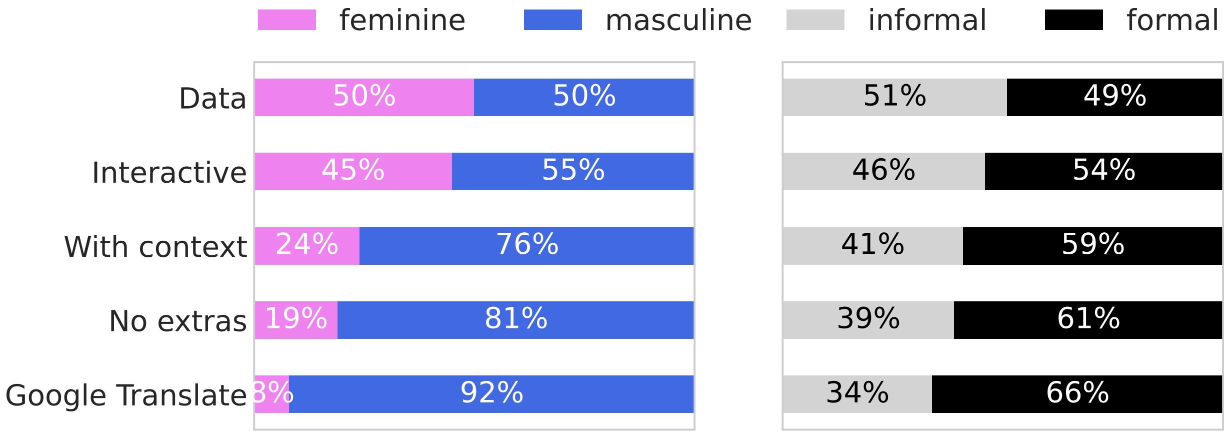


Figure 4.6 Bias in generated translations for French and Spanish on "it" resolution (left) and formality (right).

4.5.6 When is context better than interaction?

In this section, we provide analysis that describes common areas of improvement for *generalist interactive-chain prompting*.

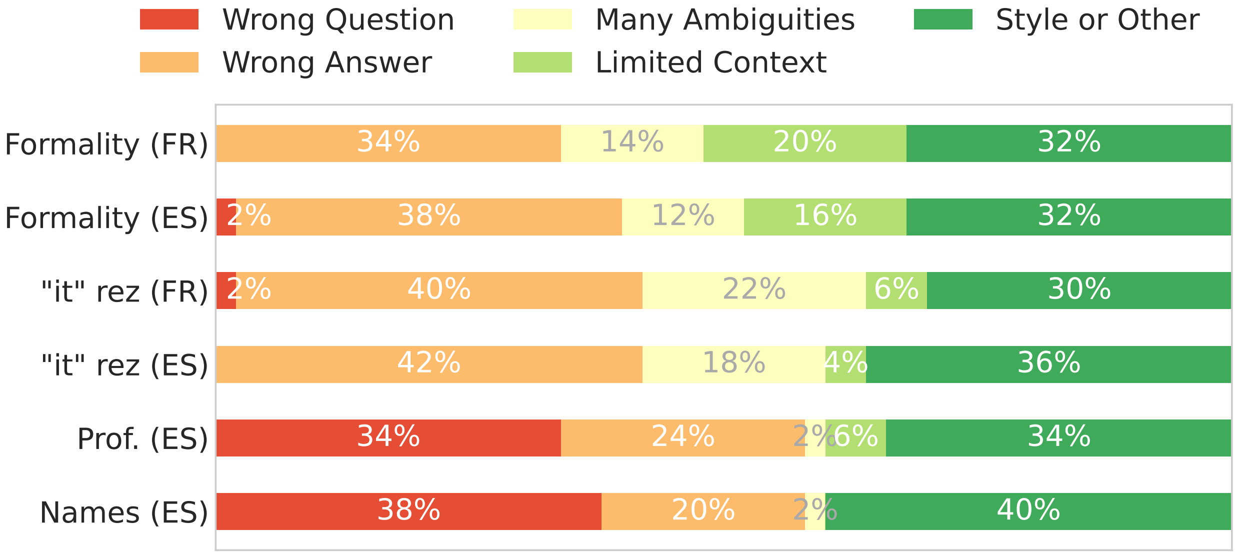


Figure 4.7 Error analysis. rez = "it" resolution, Prof. = Neutral profession, Names = Gender Neutral Names

We first isolated test samples for French and Spanish for four ambiguities (formality, "it" resolution, neutral professions and gender neutral names) where the *BLEURT* scores were less than or equal to *LLMwCxt* scores. We then randomly sampled 50 interactions and manually analysed the interaction chains (query, question, context, answer, translation).

This led us to five types of errors: (1) wrong question, when the Translator LM asked a question not related to the ambiguity; (2) wrong answer, when the User LM did not provide correctly disambiguate; (3) many ambiguities, when the query had multiple unresolved ambiguities or the User LM answer also contained ambiguities; (4) limited context, when the context was not sufficiently informative to resolve ambiguities; (5) style or other, when generated translated text had discernible differences with the ground truth.

Error Type	***en* Query (S) and Question (Q)**	**Sim User Context (C) and Answer (A)**	**Observation**
Wrong Question	**S**: But I swear to you it wasn't me. **Q**: What does "it" refer to?	**C**: I just thought that he'd blame me for predicting his death [...]. **A**: "it" is death	**S** can be translated without information on "it" and did not ask a question to disambiguate formality.
Wrong Answer	**S**: Develop it further, Leonard. **Q**: What does "it" refer to?	**C**: – Get me a complete rundown on Miller [...]. – That's a good idea. **A**: "it" is a plan	"plan" is masculine in fr and es. However, "it" refers to "idea", which is feminine in fr and es.
Many Ambiguities	**S**: If anyone asks, you're a relief worker. **Q**: Who does "you" refer to?	**C**: – Okay, so I'm going to go with you. – White girls don't do runs. **A**: 'informal' since the speaker talking to a friend "Aaron"	The answer is correct; however, the name Aaron is gender neutral and was resolved incorrectly, impacting "worker" translation.
Limited Context	**S**: I'll bring it right over. **Q**: What does "it" refer to?	**C**: – You didn't get it? – Really? – Just a second... **A**: "it" is a harp	"harp" is likely wrong. We cannot determine what "it" is from the given context.

Table 4.4 Examples of interaction chain errors.

Fig. 4.7 shows that the majority of errors are from wrong User LM answers for formality and "it" resolution. This partially confirms our hypothesis in Subsection 4.5.4. For tasks involving unseen ambiguities, the majority of errors come from the Translator LM with 68% to 78% of sample chains having the wrong question or noticeable differences in generated translated text style or form. We provide examples of interaction chains for each type of error in Table 4.4.

4.6 Conclusion

We propose *interactive-chain prompting* (*InterCPt*), a prompt-based interactive multistep computation technique that first resolves cross-lingual ambiguities in the input queries and then performs conditional text generation. In our experiments, we assume that ambiguities are subproblems and show that a question-answer interaction between a Translator LM and User LM to resolve ambiguities greatly improves generated translation quality. We have created and released a new datasets that covers five ambiguities: formality, "it" resolution, polysemy, gender neutral names and neutral professions for four different language pairs. Empirical results show that *InterCPt* outperforms other prompt-based techniques that have access to all background information and context to directly resolve ambiguities. We find that *InterCPt* MT is an emergent property of parameter scale that allows Large LMs to perform interactive generation tasks while other prompt-based techniques exhibit flattening scaling curves. *InterCPt* can be considered a step forward more effectively interacting with machine learning systems.

CHAPTER 5 ARTICLE 4: BLOCK-STATE TRANSFORMERS

Jonathan Pilault[1,2,4,*], Mahan Fathi[1,2,3,*], Orhan Firat[1],
Pierre-Luc Bacon[2,3], Ross Goroshin[1] and Christopher Pal[2,4]

Accepted in Advances in Neural Information Processing Systems 36 (NeurIPS), Publication Date: 09/21/2023

[1]Google DeepMind [2]Mila [3]Université de Montréal [4]Polytechnique Montréal

[*] Equal contribution[1].

Abstract

State space models (SSMs) have shown impressive results on tasks that require modeling long-range dependencies and efficiently scale to long sequences owing to their subquadratic runtime complexity. Originally designed for continuous signals, SSMs have shown superior performance on a plethora of tasks, in vision and audio; however, SSMs still lag Transformer performance in Language Modeling tasks. In this work, we propose a hybrid layer named Block-State Transformer (BST), that internally combines an SSM sublayer for long-range contextualization, and a Block Transformer sublayer for short-term representation of sequences. We study three different, and completely parallelizable, variants that integrate SSMs and block-wise attention. We show that our model outperforms similar Transformer-based architectures on language modeling perplexity and generalizes to longer sequences. In addition, the Block-State Transformer demonstrates more than *tenfold* increase in speed at the layer level compared to the Block-Recurrent Transformer when model parallelization is employed, yielding favorable perplexity–memory trade-offs at long context and strong performance on Long Range Arena benchmarks, under comparable parameter budgets.

[1] Jonathan Pilault a contribué dans la conception des algorithmes, l'experimentation, l'analyse des résultats, amélioration de l'efficacité du logiciel, analyse théorique, rédaction de l'article, recherche bibliographique et la réponse aux réviseurs de conférence.

5.1 Introduction

Transformers have shown impressive performance on a wide range of natural language processing (NLP) tasks. While primarily used for language modeling, the Transformer architecture [115] has also been successfully applied to other tasks outside of the NLP and have mostly replaced Recurrent Neural Networks (RNNs). Several factors contribute to this success, including computational efficiency and architectural inductive biases that are well-suited for training on natural language tasks at scale. On the computational upside, Transformers are able to process tokens of a given input sequence in parallel, making the most of modern accelerator hardware. Moreover, the attention mechanism enables Transformers to find relationships in longer sequences by providing ready access to all the extracted information from past tokens when inferring the next token. Compared to RNNs and LSTMs [202], the benefits of self-attention are two-fold: (i) the capacity of what could be stored and directly accessible as context is drastically increased, and (ii) training on longer sequences is more stable [203, 204].

Given the remarkable achievements of Transformers in language modeling tasks, and their improved performance at scale on hard NLP tasks such as reasoning and question answering [205, 206, 207], the demand for deploying even deeper and larger networks is greater than ever before. An orthogonal scaling dimension, which could be potentially even more consequential, is the size of the input sequence. Despite the several advantages of Transformers over RNNs, it is still problematic to scale the input sequence length, again for both computational performance and quality reasons. Further, the Transformer's runtime is quadratic with respect to the input sequence length, which makes training these models increasingly expensive. Furthermore, Transformers with attention, that is local [208], sparse [209, 210, 211], low-rank approximated [212] or linearized via kernel methods [213, 214], notoriously struggle on long-input classification tasks [215]. Vanilla transformers can be unstable when trained on long sequences [216] and token importance is concentrated in a local receptive field of around 50 tokens around the current time step [217].

An emerging body of research suggests that State Space Models (SSMs) can serve as an alternative to Transformers because they are able to capture dependencies in extremely long sequences, while being more computationally efficient and parallelizable [218]. While still falling into the category of autoregressive sequence models, the underlying linear time-invariant dynamical system of SSMs allows the efficient processing of sequences using parallelizable convolution operators with the Fast Fourier Transform (FFT) [219], with $\mathcal{O}(L \log L)$ complexity, where L is the length of the sequence. Moreover, retention of past information over long sequences, up to thousands of steps, can be ensured by deriving recurrent update rules by borrowing ideas from online function approximation [220, 221]. SSMs have recently outperformed Transformers on long-range dependency benchmarks by a large margin [215]. Despite their success on long-range classification tasks, SSMs have not yet completely matched Transformers as an off-the-shelf sequence model for general language modeling tasks [222].

Recent findings suggest that Transformers and SSMs are complementary models for the purpose of language modeling [13]. In this work, we propose an architecture that integrates a strong local attention-based inductive bias with the long-term context modeling abilities of SSMs into a single layer, that we call *Block-State Transformer* (BST). Our model is able to process long input sequences, while still incorporating an attention mechanism to predict next tokens. BST is fully parallelizable, scales to much longer sequences, and offers a 10× speedup compared to comparable Transformer-based layers.

In every BST layer, an SSM takes the entire sequence as input and maps it into a "context" sequence of the same length. The SSM sublayer takes advantage of FFT-based convolutions. This sequence of context is then divided into blocks of equal size, i.e. window length (W), and each context block is then fed to a Block Transformer layer, that attends to the subsequences of size W as defined in [12]. The block of input token embeddings are then cross-attended to the corresponding block of context states; see Figure 5.1. Note that by introducing SSMs as a means of contextualization, we completely remove the need for sequential recurrences and we are able to run our hybrid SSM-Transformer layer fully in parallel. The resulting runtime complexity can be expressed as the sum of $\mathcal{O}(W^2) + \mathcal{O}(L \log L)$, where the first term represents the time complexity of the Transformer sublayer, while the second term represents the time complexity of the SSM sublayer. This is a major improvement over $\mathcal{O}(LW)$ of Block-Recurrent Transformer, so long as hardware to support parallel computation is available.

Moreover, due to hardware imposed restrictions, the runtime complexity of the SSM on a full sequence is comparable to that of Block Transformer on a block of tokens, which further implies the absence of a speed bottleneck in the BST layer, empirically validated for sequences containing hundreds of thousand of tokens. This is evident by observing that the bottom-most two lines on the left of Figure 5.4 are almost overlapping.

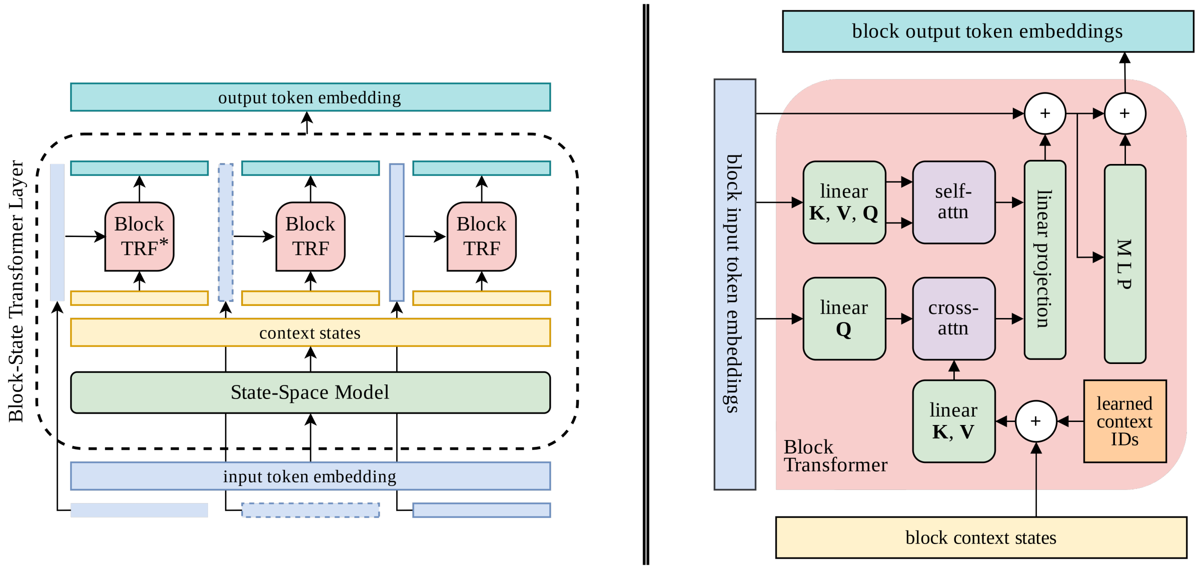


Figure 5.1 Block-State Transformer layer. The BST-SH layer is illustrated on the left, and includes a state space model (SSM, in green) and Block Transformers (in red). For demonstration purposes the sequence is divided into 3 blocks in the picture. The details of the Block Transformer sublayer are on the right. *TRF = Transformer.

5.2 Related Work

This work is primarily related to two branches of recent research: (i) combining local attention with recurrent networks in order to extend their capacity to capture long-range dependencies, beyond the length of the attention window size, and (ii) State Space Models (SSMs) which describe sequences via linear dynamical systems whose outputs can be computed in parallel. Block-Recurrent Transformer (BRecT) [12] uses a recurrent memory mechanism to extend the theoretical context length of the Transformer. In the recurrent unit of the BRecT cell, the updates made to the “recurrent state vectors,” are extracted by employing a cross-attention mechanism over a block/window of input token embeddings. Different from their work, we use linear state space models instead of recurrent cells to maintain context states. We also conduct a more extensive exploration of maintaining and updating context states. Earlier works that augment transformers with a non-differentiable external memory include the Memorizing Transformer [223]. Transformer-XL [208] was an early work that combined recurrent memory with Transformers. Our work can be seen as a continued evolution of those models incorporating state-of-the-art recurrent memory models inspired by SSMs.

State space models can be considered as linear RNNs [221]. This simplicity facilitates their analysis and even enables analytical derivation of recurrent weights for optimally representing arbitrarily long sequences. The linear property also allows the recurrence to be unrolled and parallelized during training and inference [218]. Our work combines these state-of-the art models, enabling Transformers to leverage theoretically infinite context.

Other works have attempted to replace Transformers, and their attention mechanism with SSMs [13, 224, 222, 225], however despite recent progress, the performance achieved by the Transformer architecture remains unparalleled in language. Nevertheless, SSMs are able to capture longer range dependencies than Transformers in both theory and practice, while also being highly parallelizable [219, 226]. We therefore elect to combine the best aspects of SSMs and Transformers into a single model. The idea of communication across blocks, similar to GSS [13], was later implemented by MEGA [224], through an Exponentially Moving Average (EMA) update rule instead of SSMs[2]. However, both GSS and MEGA use a single-head Gated Attention Unit (GAU) [227]. MEGA further mixes layer inputs, GAU outputs and EMA outputs via two gating mechanisms. Our method uses a simpler architecture to mix signals from local attention and SSM outputs via cross-attention, allowing us to plug any out-of-the-box SSMs or attention layers. Further, we investigate three ways to mix SSM signals with attention as outlined in Section 5.3.3.

5.3 Method

We consider the problem of next token prediction via a decoder-only language model. This seemingly simple pretext task has led to spectacular progress in language understanding [6, 205, 228]. During training, the decoder takes in a sequence of length L of tokens embeddings and is tasked to generate the next token at every step in the sequence.

We start by a brief review of SSMs that are essential for understanding the Block-State Transformer layer (5.3.1). Our full Block-State Transformer architecture is outlined in Section 5.3.2. Section 5.3.3 describes three approaches for integrating SSM states into the attention mechanism. Important implementation details are described in Section 5.3.4.

5.3.1 State Space Preliminaries

State space models can be divided into two categories:

State Spaces: Structured Kernels S4 [218], S5 [229], S4D [230], DSS [231], follow a structured initialization of the convolutional kernel by unrolling a linear time-invariant (LTI) dynamical system of the following form:

$$\begin{aligned} x_k &= \mathbf{A}x_{k-1} + \mathbf{B}u_k \ , \\ y_k &= \mathbf{C}x_k + \mathbf{D}u_k \ . \end{aligned} \tag{5.1}$$

[2] The authors in [224] show a mathematical form of EMA that has a state transition and also derive a convolution kernel to efficiently compute EMA similarly to S4.

The system is parameterized by a state matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, vectors $\mathbf{B} \in \mathbb{R}^{N \times 1}$, $\mathbf{C} \in \mathbb{R}^{1 \times N}$, and $\mathbf{D} \in \mathbb{R}^{1 \times 1}$, the SSM maps a 1-D input signal u_k, to a 1-D output signal y_k. Internally, the SSM projects the input signal to an N-D representation state x_k, before mapping it down to a scalar using the $\mathbf{C}$ vector. The term $\mathbf{D}u_k$ can be thought of as a skip connection and will be omitted for the remainder of the discussion for convenience. The output of the above recurrent equation, y_k, can be computed as a discrete convolution, by realizing that the recurrence can be explicitly unrolled:

$$\begin{aligned} \text{Let} \quad x_{-1} &:= \vec{0} \ , \\ y_k &= \sum_{j=0}^{k} \mathbf{C}\mathbf{A}^j\mathbf{B} \cdot u_{k-j} \ . \end{aligned} \tag{5.2}$$

The $\mathbf{C}\mathbf{A}^k\mathbf{B}$ entries are collected to create the SSM kernel $\mathbf{K} \in \mathbb{R}^L$, and the convolution could be expressed as:

$$\begin{aligned} \mathbf{K} &= (\mathbf{CB}, \mathbf{CAB}, \ldots, \mathbf{CA}^{L-1}\mathbf{B}) \ , \\ y_k &= \sum_{j=0}^{k} \mathbf{K}_j \cdot u_{k-j} \ , \quad y \ = \ \mathbf{K} * u \ . \end{aligned} \tag{5.3}$$

Given an input sequence $u \in \mathbb{R}^L$, it is possible to compute the output $y \in \mathbb{R}^L$ sequentially through the recurrence in Equation equation 5.1. While this property is useful for autoregressive decoding, sequential computation is prohibitively slow to train with long inputs and, instead, the convolution from the Equation equation 5.3 can be used to compute all elements of y in parallel. This is done via Fast Fourier Transform (FFT) [219], provided we have already computed $\mathbf{K}$.

Additional inductive biases have been imposed on SSMs by *analytically deriving* closed-form expressions for the matrices $\mathbf{A}$ and $\mathbf{B}$ using the HiPPO framework [221]. In this framework, the state x_t represents the coefficients of polynomials that approximate the sequence u_t.

Explicitly Parameterized Filters In contrast to structured kernels, one can parameterize the convolution kernel, as trainable weights and optimize them, $\bar{\mathbf{K}} \in \mathbb{R}^L$. However, this would result in poor performance unless certain types of regularization are applied to the kernel. [226] simply makes use of squashing the kernel weights, and subsequently applying a smoothing technique. Trainable kernels are also used in attention-free alternative models to Transformers, such as Hyena [225], which involves exponentially decaying the weights along the kernel:

$$\bar{\mathbf{K}}_t \quad = \quad e^{-\alpha t} \cdot \Big(\texttt{FFN} \circ \texttt{PositionalEncoding}\Big)(t) \ , \tag{5.4}$$

where $\bar{\mathbf{K}}_t$ is an entry in the filter at location t, and FFN is a feed-forward network used for decoupling the parameter count from the seuqnece length.

5.3.2 Block-State Transformer (BST) Layer

We now introduce the Block-State Transformer layer, which combines SSMs with Block Transformers. At each training iteration, a sequence of L tokens, is sampled from a longer document. The tokens are then embedded and fed to the model. Our model consists of a stack of Block-State Transformer layers. Each BST layer optionally includes an SSM sublayer that is responsible for providing long-range context to the Block Transformer layer, which operate similarly to a Block-Recurrent Transformer (BRecT) cell. The SSM sublayer takes the sequence of token embeddings from the previous layer as input, and produces a sequence of the same length L as the output.

The output of the SSM is contextually encoded, meaning that entries at every time-step, potentially include information about all the time steps preceding elements in the sequence. We collect a number of "context states," S, from the context sequence, and we set $S \ll L$. In order to prevent the model from accessing future information, we only allow the model to access context states that precede the current token. Various ways to gather context states from the context sequence are discussed in section 5.3.3 in detail.

The context states are fed to the Block Transformer, in place of what was referred to as "recurrent state vectors" in Block-Recurrent Transformer [12]. The subsequent operations, shown on the right side of Figure 5.1, are kept unaltered, except that we no longer need to run the recurrent unit of the BRecT cell since we are maintaining the context via an SSM. In addition to the context states, the Block Transformer also receives a block/window of length W of token embeddings as input, which are cross-attended to the context states. The output of the cross-attention operation is then concatenated with that of self-attention over the input embeddings, followed by a simple projection.

In addition to the ability of SSMs to retain information over longer time horizons compared to Transformers and RNNs, using the SSM to maintain context states as a replacement for recurrent cells makes for a more computationally efficient layer. Removing recurrence by integrating SSMs into Transformer layers, allows the Block-State Transformer layer to be fully parallelizable, whereas the Block-Recurrent architecture processes blocks of tokens sequentially using a for loop.

5.3.3 Context States

Although the latest SSM output technically contains information about the entire sequence, retrieving individual tokens from only the final state may not be feasible. To compensate, we concatenate a sequence of states, corresponding to the latest block of tokens. This is also analogous to the approach taken by BRecT. This representation ensures *retrievability* and ease of access, through *redundancy*. It is redundant because adjacent states are highly correlated, however this also makes it possible to easily recover the current block of tokens, if necessary.

In our approach, the context states are constructed from the output of the SSM and fed to the attention heads of the Transformer. These context states can be constructed in various ways. To guide these design decisions we consider each of the below proposed schemes as introducing *retrievability* at the cost of *redundancy*. The shape of the output of a single SSM layer is $(B \times L \times D)$, where B is the batch size, L is the number of the tokens processed, and D is the embedding dimension. When doing cross-attention in the Transformer cell with H different heads, this tensor needs to be transformed into a context tensor of shape $(B \times S \times D \times H)$, where S is the number of context states; we usually set $S \ll L$ and $S = W$ similar to Block-Recurrent Transformers (BRecT).

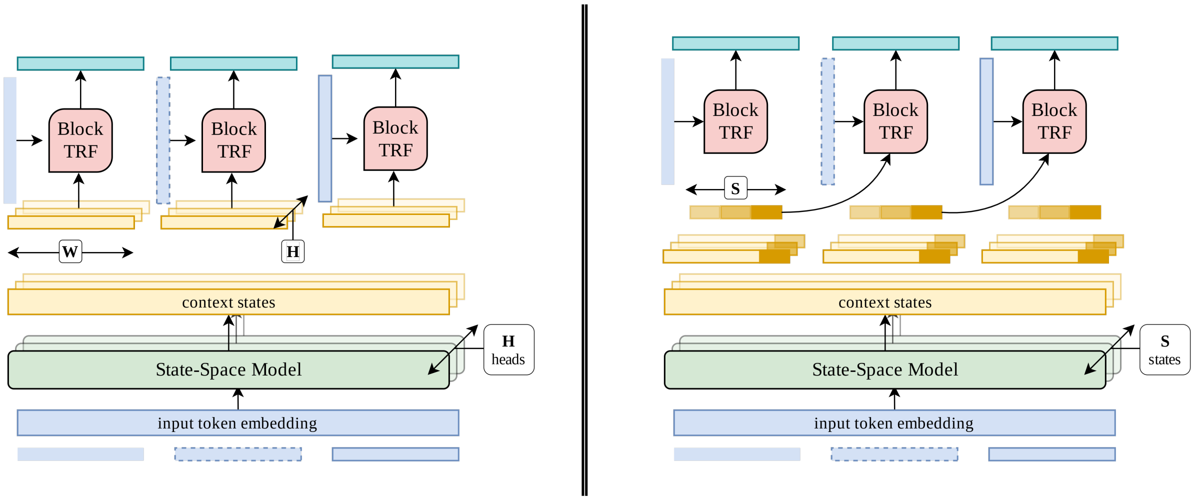


Figure 5.2 Summarizing our approaches. The left side shows the cases where the SSM is required to output Multi-Head (**MH**) contexts. On the right Multi-Filter (**MF**) approach is depicted where the last entries from the previous window are concatenated into a set of context states of size S. Dashed lines represent the current block.

We now discuss the three different approaches that we evaluate to generate a context tensor for each block sequence:

SH: Single-Head The first approach constructs the context tensor by sequentially concatenating the S states from the SSM with a single filter (each of size D). Note that because the SSM captures information from preceding blocks, the context state also captures information about blocks that preceded the current block. The resulting context vector is highly *retrievable* and *redundant*, as defined above. As in typical Transformers, fully connected layers are used to project each context vector to H different heads of size D. Note that in the cross-attention operation, context states that correspond to future tokens from the current block need to be causally masked out. In this case we set $S = W$, and we pick the window of SSM outputs that correspond to the current block, and a triangular mask is used to implement causal masking of context states. This approach is shown in Figure 5.1.

MH: Multi-Head This approach differs from Single-Head (SH) in that here the SSM is tasked to generate a separate output for different heads. We use separate $[\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_H]$ matrices, to produce context states that are fed to the attention heads. This enables the SSM to extract complementary features from the summarized history. The conceptual difference is that the $\mathbf{C}$ matrix, from Equation equation 5.1, has direct access to the full memory state of the SSM (x_k), that in theory could be thought of as a compact representation of the history, before it gets mapped down to a scalar. The Multi-Head (MH) approach is illustrated on the left side of Figure 5.2. Because the H different $\mathbf{C}$ matrices may extract complementary information, the context vector constructed by this method is theoretically less *redundant* compared to the single-head method described above.

MF: Multi-Filter In this approach the SSM sublayer produces S context states, which we set to be independent from W. This is done by convolving the sequence of embeddings with S different kernels/filters. The output of each convolution operation, corresponding to a specific filter, is a tensor of shape $(B \times L \times D)$. After convolving the input with all the filters, the context states of size D that correspond to the *last token* from the previous window are stacked together to make a $(B \times S \times D)$ tensor. Feed forward networks are then used to lift this tensor to different heads, $(B \times S \times D \times H)$. Different from the previous two approaches, the context is formed by taking only the *last* S context states, from the previous window, outputted by the S SSMs. The context is less redundant because it no longer consists of adjacent SSM states. Since the context is taken from the entries of the previous window, cross-attention masking is no longer required, as shown on the right of Figure 5.2.

The memory states of the Multi-Filter (MF) approach is least *redundant*, while Multi-Head (MH) strikes a middle ground, and Single-Head (SH) has the most redundancy. The incorporation of *redundancy* in these approaches aims to facilitate *retrievability* of the most recent context captured by the SSM, albeit at the expense of potentially inefficient *utilization* of the network capacity. The last approach attains highest *utilization*, as the cross-attention is done in the space of unique features extracted by specialized filters.

5.3.4 Implementation Details

Context IDs & Positional Embedding To allow distinction between the entries supplied to the attention mechanism, a positional embedding is commonly added to the inputs. When using the Multi-Filter (MF) approach, the collected context states correspond to different features extracted from the sequence, hence we add a set of unique learned "context IDs" to the context states, before using them as input to cross-attention. However, in the cases where the context states correspond to different time-steps along the sequence, namely Single-Head (SH) and Multi-Head (MH) approaches, inherent positional encoding is incorporated into the context states, due to the incremental nature of convolutions; as such, we find the addition of context IDs to be unnecessary. We also realize that we do not need to add global positional bias to the token embeddings, and use a T5-style relative position bias [141] instead, as the SSM does also encode positional information into the context.

Down-sampling Consistent with findings in [13], we find FFT operations to be the main source of bottleneck when training SSMs on TPUs. We project the input embeddings to a lower-dimensional space, that is a quarter of embedding size in our experiments, this reduces the required total number of FFTs by a factor of 4. The output of the SSM, i.e. the context states, are later lifted to the original embedding size before being passed to the Block Transformer.

5.4 Results

Our results are presented in Table 5.1. We conduct experiments with BST on three different datasets, PG19, arXiv and GitHub, allowing us to test our method on a suite of varying documents lengths composed of English texts, latex scientific articles and source code.

PG19 dataset is from a large collection of full-length books from Project Gutenberg [232]. All extracted 28,602 books were published prior to 1919 and contain 6,966,499 English language words. When tokenized, each PG19 book has between 50k-100k tokens. PG19 has become a popular benchmark for measuring progress on long-range language modeling performance. We report the "test" split evaluation performance.

arXiv dataset is a corpus containing scientific and technical articles on the subject of Mathematics [223]. The arXiv dataset contains latex source code as well as items such as theorems, citations, definitions that are referenced and discussed over long ranges of text. Using the same vocabulary as in [223] and [12] for a fair comparison, many special characters are broken up into small subwords. As a result, the number of tokens per paper in the arXiv dataset is approximately equal to the number of tokens per book in PG19. We report perplexity on "test" split.

GitHub dataset [223] is the largest of the three datasets and was assembled by extracting GitHub code repositories with open-source licences. Files were filtered to only contain the following programming languages: C, C++, Java, Python, Go and Typescript. While code files are relatively small, there are many import dependencies between each file. By traversing the directory tree and concatenating all code files along the path, a single document that preserves a repository's structure and dependencies is created. We report performance on the "validation" split.

For a fair comparison with the baselines, we keep the vocabularies consistent as used by [12] and [13]. Specifically, we used a pretrained T5 vocab with 32k tokens for PG19 [233] and LaMDA vocab with 32k tokens [206] for both arXiv and GitHub datasets. Due to the long training times and large number of experiments, we only provide error bars for the PG19 ∼200M parameter models by running our models with three different random seeds. BRecT:fixed:skip error bars are from [12].

5.4.1 Comparing our Baselines and Models

We experiment three different types Block-State Transformer (BST) models: BST-SH, BST-MH and BST-MF as described in Section 5.3.3. Our models do not use global learned positional embeddings but encode positional awareness with an SSM at the first layer, right after the word embedding layer. We organize models into two groups: (i) *fixed window size* have either a 512 or a 2048 token training window size; and (ii) *fixed parameter count* have either a ∼200M or ∼400M total parameters. We run experiments with two types of SSMs:

BST:{SH,MH,MF}:S4 encode long context using a Structured State Space Model (S4) [231]. As described in Equation equation 5.3, S4 kernel matrix **K** is compiled from matrices **A**, **B** and **C** and is independent of the length of the input evaluation sequence length. We show that the structured parameterization of **K** allows our BST models to generalize to longer

Table 5.1 Perplexity of each model. The results for XL:2048, Slide:12L and BRecT:fixed:skip are from [12] by converting $\log_2$ of perplexity to raw perplexity. GSS-Hybrid-L performance was taken from [13]. Results with ± are average scores and error bars of runs with three different random seeds. For a *smaller computational budget*, BST provides a small perplexity improvement compared to BRecT on PG19 and GitHub. For the same computational budget, BST outperforms GSS-Hybrid-L across datasets by 1.5% to 4%.

Model	eval seq.	window	number	TPUv4 hours (k)	PG19	arXiv	GitHub
	length	length	params	PG19/arXiv/GitHub			
Slide:12L	4096	512	190M	0.5 / 0.5 / 1.8	12.12	2.69	2.28
Trsf-XL:2048	2048	2048	190M	0.8 / 0.8 / 3.0	11.96	2.48	2.01
BRecT:fixed:skip	4096	512	196M	0.8 / 0.8 / 3.0	11.55 ±1.1	**2.36**	2.04
BST:SH:S4			202M	0.5 / 0.5 / 1.8	11.57 ±1.1	2.51	2.14
BST:MH:S4			218M	0.8 / 0.8 / 1.8	11.60 ±1.1	2.52	2.15
BST:MF:S4			217M	0.5 / 0.5 / 1.8	11.63 ±1.2	2.48	2.07
BST:SH:unstruct			206M	0.5 / 0.5 / 1.8	**11.52** ±1.1	2.49	2.09
BST:MF:unstruct			221M	0.5 / 0.5 / 1.8	11.56 ±1.2	2.44	**2.03**
GSS-Hybrid-L	4096	512	373M	0.8 / 0.8 / 1.8	10.52	2.51	1.88
BST:SH:S4-L			366M	0.8 / 0.8 / 1.8	10.47	2.49	1.86
BST:MF:S4-L			383M	0.8 / 0.8 / 1.8	10.52	2.46	1.84
BST:SH:unstruct-L			371M	0.8 / 0.8 / 1.8	**10.37**	2.46	1.85
BST:MF:unstruct-L			388M	0.8 / 0.8 / 1.8	10.42	2.41	**1.83**

lengths. We refer the reader to section 5.4.2 for results on length generalization. We only run one BST:MH using S4 since the model requires 8% more parameters while performing on par with the faster BST:SH variant. BST:MF also has 8% more parameters but performs better on arXiv and GitHub compared to SH. Interestingly, SH performs better than MF on the PG19, a dataset where local context is more important to predict the next token compared to arXiv and GitHub. We posit that this is likely due to the ability of the SH model to *retrieve* the most recent context captured by the SSM.

BST:{SH,MF}:UNSTRUCT are based of unstructured parameterized convolution filters, inspired by the Hyena Hierarchies [225] convolutional kernel. We exclude the utilization of the multiplicative gating mechanism employed in Hyena Hierarchies and solely apply the regularizations implemented on the parameterized kernel, denoted as $\bar{\mathbf{K}}$ in Equation equation 5.4. This formulation has two important advantages over S4: (1) the $\bar{\mathbf{K}}$ kernel does not need to be recompiled, allowing speedups when using multiple filters; (2) $\bar{\mathbf{K}}$ has more free parameters because it is no longer restricted by $\mathbf{A}$, $\mathbf{B}$ matrices in equation 5.3, potentially providing richer representations that can explain the improved perplexity scores over S4 variants. Nonetheless, UNSTRUCT kernel $\bar{\mathbf{K}}$ relies on learned positional encoding which makes the method less extendable to larger length sequences at inference..

We compare the Block-State Transformer to four different baselines:

TRSF-XL:2048 [208] is a Transformer with a training window size of 2048. As expected, increasing the window size improves perplexity, especially on the arXiv and GitHub datasets. However, this model performs worse than BST:SH:HYENA on PG19 and is much slower, bottlenecked by the attention layer on higher sequence lengths.

SLIDE:12L [12] This model is almost identical to TRSF-XL:2048. It uses however a sliding window of size 512 over a segment of 4096 tokens. The sliding window is differentiable over two blocks, while TRSF-XL does not backpropagate through the cached keys and values from the previous window. This simple baseline is closest in terms of training speed to BST:SH. The perplexity scores show that integrating a representation of the past, as with BRECT and BST, positively impacts LM performance.

BRECT:FIXED:SKIP [12] is the strongest performing and fastest Block-Recurrent Transformer architecture in [12]. This architecture is very similar to SLIDE:12L. There is however a sequential recurrent “skip” configuration, a simple linear layer gating mechanism that combines current block hidden representation with past information from the previous blocks.

GSS-Hybrid-L [13] is the closest SSM-Transformer hybrid model that was tested on long-range language modeling tasks. GSS-Hybrid-L is based on the Diagonal State Space (DSS) [231]. DSS and S4 are similar in performance and architecture, only differing on the initialization of the kernel $\mathbf{K}$ [230]. [231] further improves on DSS for LM tasks by introducing a Gated State Space version called GSS, which performs better on PG19, arXiv and GitHub. Unlike our method, GSS-Hybrid-L does not directly integrate SSMs states into the attention mechanism but only interleaves 32 GSS layers with Transformer layers. It must be noted that the GSS-Hybrid-L scores were obtained after grid searching over four learning rates $\{6.4,3.2,1.6,0.8\}\times 10^{-3}$ and used a different learning rate and weight decay for the SSM layer and the Transformer layer to avoid training instabilities. In our experiment, we did not use grid search and used the same learning rate for all layers. BST results demonstrate that integrating SSM states into the Transformer attention provides larger benefits than interleaving SSM and attention layers as in GSS-Hybrid-L.

Fixed compute budget. As seen in Table 5.1, we track the exact amount of compute in TPUv4 hours that was spent training each model. The training TPUv4 hours for Slide:12L, Trsf-XL:2048, BRecT:fixed:skip and GSS-Hybrid-L were taken from [13]. The TPUv4 hours metric measures the compute cost of training models. For our experiments, we align our training times with GSS-Hybrid-L for a fair comparison. Smaller parameter models all have 12 layers, 8 heads of size 128, embedding vectors of size 1024, an MLP with a hidden layer size of 4096 with ReLU activation functions. For larger BST models, we double the intermediate layer size from 4096 to 8192 and increase the number of attention heads to 12.

Training details We use the same training setup as [12] and we perform our experiments using the Meliad library[3] in JAX/Flax [234, 235]. We use the Adam optimizer [236] and a batch size of 32 and a sequence length L of 4k for training. Using a structured SSM's recurrence (such as S4) in the first layer allows us to extend the positional encoding to various lengths at inference. Smaller BST models have Block-State layer integrated in Transformer layers {1, 7, 9} and larger BST models at layers {1, 5, 7, 9}. Since our datasets contain long documents, it is possible to train on larger sequence lengths L. Training on 4k sequence lengths allows us to test length generalization since the convolution kernel $\mathbf{K}$ in Equation equation 5.3 can be extended to any sequence length L. However, since we show in Section 5.4.2 that our model works well when extended to unseen lengths, we did not find it necessary to run expensive experiments with higher sequence lengths. For the MF model variants, we lower the SSM state dimension D by an additional factor of two to improve

[3] https://github.com/google-research/meliad

FFT efficiency. The state dimension reduction has negligible impact to perplexity. The MF models have $S = 32$ filters while the larger MF models have $S = 64$ filters.

5.4.2 Evaluating Length Generalization capabilities

We present our length generalization analysis and report perplexity in Figure 5.3. Our models and baselines all have ∼400M parameters, are trained on a sequence length of 4k and tested on sequences with *lower* and *higher* sequence lengths of {512, 16k, 65k}.

We notice that all models have similar perplexity for sequence lengths of 512. Both BST:SH:S4-L and GSS-Hybrid-L generalize well on 16k and 65k sequence lengths for PG19 and GitHub. For arXiv, GSS-Hybrid-L and BST:MF:unstruct-L perplexities increase drastically, potentially due to noise in the arXiv dataset (as indicated by variation in perplexity metric over time). [13] also reported that larger GSS models had difficulty generalizing to higher lengths. Interestingly, for arXiv again, BRecT:fixed:skip-L performs very well at higher sequence lengths. We hypothesize that the Block-Recurrent model's access to the entire past during training, via a non-differentiable cache of representations across sequences, helps retain a "memory" of dependencies between key items in an arXiv article allowing the model to access past symbols, definitions, theorems or equations beyond the 4k training sequence length. We also note that BST:MF:unstruct-L and BRecT:fixed:skip-L outperform other methods on PG19 up to a sequence length of 16K. Perplexity performance on PG19 is perhaps less reliant on long term relationships between tokens, which can explain the performance of models that have no explicit built-in mechanisms for length generalization.

The analysis also allows us to draw a clear distinction between *structured* and *unstructured* SSMs integrated in hybrid architectures. As previously mentioned in Section 5.3.1, SSMs such as DSS and S4 use a structured kernel $\mathbf{K}$, built from learned matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ for any sequence length L in Equation 5.3. Since $\mathbf{K}$ is extendable to any arbitrary sequence length L, both BST:SH:S4-L and GSS-Hybrid-L have a build-in mechanism for length generalization that the unstructured BST:MF:unstruct-L model does not. BST:MF:unstruct-L performs best on the training sequence of 4K and is on-par for 512 with perplexity increasing for unseen 16K and 65K sequence lengths. **BST:SH:S4-L has by far the best perplexity for 65K sequence lengths on PG19, GitHub and arXiv.** Similarly to [12], we also notice that perplexity improves when we extend context window (sequence length) for PG19 and GitHub.

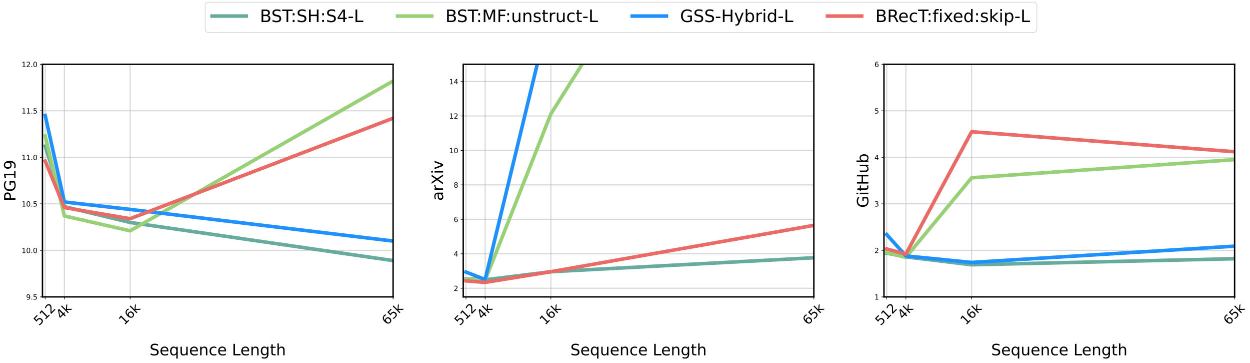


Figure 5.3 Length Generalization for sequence lengths {512, 16k, 65k} on PG19 (left), arXiv (middle) and GitHub (right). BST:SH:S4-L generalizes better than other baselines, including GSS-Hybrid-L that uses GSS, a structured SSM. GSS-Hybrid-L numbers are from [13].

5.4.3 Efficiency

The improvement over Block-Recurrent Transformers, with time complexity of $\mathcal{O}((W^2 + S^2 + 2SW) \cdot L/W) \approx \mathcal{O}(L \cdot W)$, follows from the ability to run the Block Transformer's cells in parallel. The time complexity of the Block-State Transformer layer is comprised of the time complexity of the state space model sublayer, $\mathcal{O}(D \cdot L \log L)$, in addition to the time complexity required to execute the Transformer over the given context chunks (blocks) in parallel, $\mathcal{O}(W^2)$.

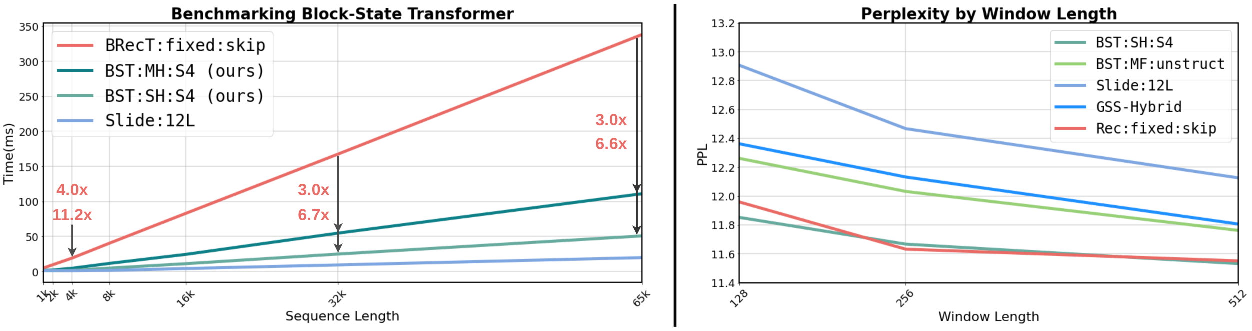


Figure 5.4 **Left**: The forward-pass computation time of a BST layer is compared against a layer of BRecT and Slide:12L. These experiments were executed on GPU, to demonstrate and exploit the parallelizability of BST layers. BST:SH is 6-11× faster than BRecT while BST:MH is 3-4× faster. **Right**: Perplexity of the trained models using different window lengths. The figure shows that increasing the training window length results, as expected, in better perplexity scores. We find however that both BST:MF:Hyena and BRecT:fixed:skip are the least impacted by decreasing window lengths.

In spite of the superlinear growth of the SSM sublayer, our experiments indicate that significant performance improvements, up to a factor of 6, remain evident for sequences as long as 65k tokens, the point at which hardware saturation began to occur. When using a structured SSM, the computational complexity is closely tied to the internal memory state size of the SSM, N – specifics may vary depending on the exact type of the SSM. We set $N = 16$ when reporting performance. Left side of Figure 5.4 shows the results of benchmarking the forward-pass of a Block-State Transformer layer on GPU. Our proposed layer runs almost 6-11× faster than Block-Recurrent Transformers (including recurrent units), and yields comparable performance to a Slide:12L layer, i.e. BRecT without the recurrence. At 4k sequence length, which is mostly used during training, BRecT layer runs almost 15× slower than Slide:12L with the same window size. We manage to reduce this gap to less than 2× with BST layer. To reflect a realistic model, for these experiments we use a fixed window length of 128, an internal state size of 16 for the SSM, and 16 heads. Moreover, to highlight the performance gains that are only due to parallelization made possible by our framework, we use same embedding size as input to the SSM, which is 512. Note that we use the vanilla implementation of FFT and inverse FFT operations provided by JAX [234]. However, we believe that the speed of our method can be further improved with recent and faster hardware-specific I/O-aware implementations introduced in other auto-diff frameworks.

5.5 Ablation Studies

In the following section, we perform ablations to investigate (1) the placement of a *single* SSM layer in Table 5.2 in the overall architecture, (2) the effects of the number of SSM layers added in Table 5.3, and (3) the size D of the SSM state in Table 5.4. For the ablations, we use the ∼200M parameter BST:SH:S4, since it is the fastest model, and assess various configurations on PG19.

Table 5.2 A single BST at various layer index.

Layer idx	PPL
3	12.41
7	11.92
9	11.88
12	12.03

Table 5.3 Multiple BST layers at various locations.

# layers	PPL
2	11.69
3	11.57
4	11.21
5	11.20

Table 5.4 Increasing BST's S4 model state size D.

State Size	PPL	Step Time
8	11.95	×0.7
16	11.57	×1.0
32	11.55	×1.8
64	11.54	×3.2

In Table 5.2, we experiment adding a single BST layer at layer indices $3, 6, 9, 12$. We notice that a single BST layer with state size $D = 16$ located closer to the middle of the whole

Block Transformer stack, at index = 9, has the greatest effect on perplexity. This finding is inline with findings in prior work [223, 12].

In Table 5.3, we test if adding multiple BST layers yields improvements on performance. We start with BST layers with state size $D = 16$ at indices $0, 9$. We follow by adding another BST layer at index 7 for a total of three BST layers and then another at index 5, followed by another at index 12. Adding more BST layers lowers perplexity. However, the results seem to plateau at 5 BST layers. We note also that there is a 3.5% training step time increase for each added layer.

In Table 5.4, we train our models with different state sizes D. For the state size ablation, we use three BST layers at indices $0, 7, 9$. We find that increasing D improves perplexity to the detriment of training speed (step time). For this reason, we chose $D = 16$ for Table 5.1 BST results.

5.5.1 Scaling Experiments

In this section, we compare how BST scales compared to Transformer-XL with 4× the window size and BRecT. In Figure 5.5, we see that at lower scales, from 80M to 200M, BRecT and BST have very similar performances. Beyond 200M, the perplexity performance percentage gap between BRecT and BST increases from 2.5% at 200M paramaters to 4.0% at 1.3B parameters. The perplexity performance percentage gap between BRecT and Trsf-XL is even more pronounced as it starts at 7.6% at 200M parameters to 10.6% at 1.3B parameters.

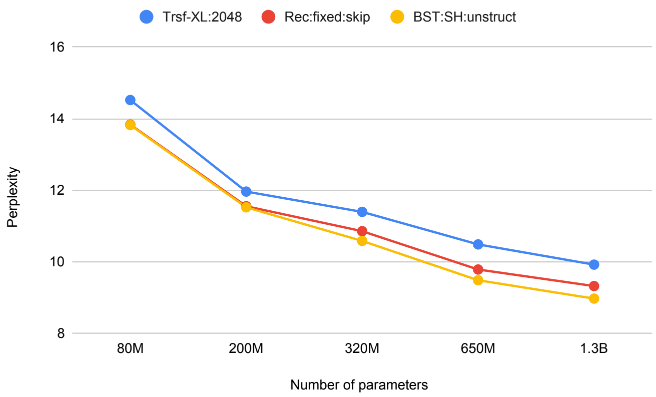


Figure 5.5
Scaling properties on PG-19.

Yellow: (BST:SH:unstruct)
12-layer Block-State Transformer.

Red: (Rec:fixed:skip)
12-layer Block-Recurrent Transformer.

Blue: (Trsf-XL-2048)
13-layer Transformer-XL.

5.5.2 Long Range Arena Experiments

While the main focus of our research was to demonstrate that hybrid Transformer-SSM models are efficient and perform well on long context autoregressive LM, we also evaluate our method on standard classification task where long range dependencies in a sequence are important to capture. In Table 5.5, we present our results on the Long Range Arena (LRA) benchmark [237] which incorporates three different modalities including text, images, and mathematical expressions. The LRA dataset also tests models on various sequence lengths from 1K to 16K.

BST:SH:S4 is composed of four BST layers (no BRT layers are interleaved) and two S4 layers on top. We use the same standard block length of 512 for BST and BRT. However, we train BST and BRT on the full sequences (up to 16K for Path-X). We use AdamW as our optimizer [238] with a warmup for the learning rate, where we start from a value of $1e^{-7}$ and increase the learning rate linearly up a specified value $\in \{1e^{-3}, 2e^{-3}, 4e^{-3}\}$ for the first 10% of training. This is followed by cosine annealing for the rest of training down to a value of $1e^{-7}$. All layers are bidirectional, including the S4 layer in BST:SH:S4 as described in [239]. Our weight decay is chosen from {0, 0.05, 0.1, 0.15} and our dropout is chosen from {0, 0.1}. Except for Path-X experiments, we use weight decays $\in \{0.03, 0.05, 0.07\}$ for all parameters except S4D matrices A and B. Also, for Path-X, the initialization range of our discretization time step Δ for PathX is decreased from (Δ_{min}, Δ_{max}) = (0.001, 0.1) to (Δ_{min}, Δ_{max}) = (0.0001, 0.01).

Our results on LRA are very promising and show that, compared to other state-of the art methods that chunk sequences into blocks, BST is able to model long range dependencies. For example, BST outperforms Mega-chunk [224] on four out of six LRA tasks and by 1.5% on the average score. However, BST still needs to improve (perhaps by extending the block size) to catch up to Mega (without chunks).

5.6 Conclusion

We have introduced a model that combines the attention mechanism of Transformers with the long-range memory mechanism, and parallelism afforded by State Space Models. We explored several memory state variants that make different trade-offs between *redundancy* and *retrievability*. Experiments show that our model can minimize perplexity on par with

Model	ListOpts	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Linear Trans.	16.13	65.90	53.09	42.34	75.30	✗	50.46
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.56
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.18
BigBird	36.05	64.02	59.29	40.83	74.87	✗	54.17
Mega	**63.14**	**90.43**	91.25	**90.44**	**96.01**	97.98	**88.21**
S4D	60.47	86.18	89.46	88.19	93.06	91.95	84.89
S4	59.60	86.82	90.90	88.65	94.20	96.35	86.09
S5	62.15	89.32	**91.40**	88.00	95.33	**98.58**	87.46
Methods with chunked input sequences							
BRecT:fixed:skip	37.29	66.14	58.76	50.41	76.33	75.89	60.80
Mega-chunk	58.76	**90.19**	**90.97**	85.80	94.41	93.81	85.66
BST:SH:S4 (ours)	**61.49**	87.63	90.51	**91.07**	**95.75**	**95.28**	**86.96**

Table 5.5 Performance on Long-Range Arena (LRA). For a fair comparison, we adjust the number of layers and model dimensions on each task so that BST and BRecT have similar number of parameters with S4 and Mega-chunk. BRecT results are from our own runs and all other baselines are from published results.

and often improves upon recent competing baselines, while achieving up to more than 10× speedups at the layer level, provided there is hardware support to fully take advantage of parallelism. This is an appealing property for scaling up BST which makes the addition of SSMs into Transformers computationally appealing. We show that integrating SSM states into the Transformer attention provides larger benefits than simply interleaving SSM and attention layers. Finally, we show that the model generalizes to longer sequences than it was trained.

CHAPTER 6 CONCLUSION

The work in this dissertation by article follows a common thread: structure the problem so that model capacity is spent where it matters, moving from input structuring to parameter structuring, to brief user–model interaction before generation, and finally to layer-level structure for long-range context. Rather than proposing a single master algorithm, the contributions show how targeted mechanisms for selection, conditioning, interaction, and hybridization address different leverage points in the LLM pipeline, providing building blocks that can be combined in future work. This chapter is organised as follows. Section 6.1 summarizes the four articles through this lens. Section 6.2 situates these ideas in 2024–2025 developments that parallel or extend them. Section 6.3 looks forward, outlining a vision for the next phase of LLMs driven by efficiency, and future research directions.

6.1 Summary of Works

Across the articles in the thesis, four principles recur. We first make important choices explicit before decoding, by selecting what the model sees such that generation emphasizes abstraction over copying on long documents. Then, we adapt compactly under constraints by routing variation through small task-conditioned adapters and allocating updates by dataset size and predictive uncertainty, achieving strong transfer with few trainable parameters and limited additional data. We also clarify intent before decoding through brief interaction that elicits missing user preferences, steering the model to the intended output on the first pass and reducing avoidable retries. Finally, we structure for long-range context via hybrid layers that couple state-space propagation with block-wise attention, yielding favorable perplexity memory trade-offs and strong Long Range Arena performance while tracking wall-clock and memory alongside accuracy.

Long-document summarization via selective conditioning via text extraction. In Chapter 2, we introduce a lightweight selection step in front of a Transformer generator to shape the conditioning context for long documents. By training the selector and generator to work in tandem, the system encourages genuine abstraction instead of surface copying. Across evaluations on arXiv, PubMed, Newsroom, and BigPatent, we observe competitive ROUGE alongside reduced n-gram copying and positive human judgments on informativeness and coherence. The key takeaway is that structuring the input with a compact set of extracted sentences can improve summary quality without increasing model size.

Parameter-efficient multi-task learning with conditional adapters and selective training data sampling. In Chapter 3, we keep most of the backbone frozen and route task variation through small, task-conditioned adapter modules. Training steps are allocated with an explicit policy that balances dataset size (temperature-scaled) and predictive uncertainty (uncertainty-weighted), all under controlled update budgets. This approach improves transfer on standard text classification and language understanding benchmarks while using few trainable parameters and bounded parameter updates, reducing storage and interference relative to full fine-tuning.

Ambiguity-aware generation through brief pre-decoding interaction. In Chapter 4, we address the inherent ambiguity that arises when user queries omit crucial context. The method reframes generation as the last step of a short interaction that asks targeted questions to surface missing preferences (e.g., formality, gender realization, pronoun resolution). Under a fixed pre-decoding budget, conditioning on these clarifications improves ambiguity resolution and downstream quality on cross-lingual evaluations, demonstrating that clarifying intent before decoding can steer outputs without lengthening generation.

Block-State Transformers for long-range language modeling. In Chapter 5, we propose Block-State Transformers, a hybrid layer that composes a state-space branch for efficient long-horizon propagation with block-wise attention for local, content-aware mixing. The design attains favorable perplexity–memory trade-offs at long context and strong performance on Long Range Arena benchmarks, offering a fully parallelizable alternative to recurrent schemes while retaining effective long-range signal flow. The result is a structural mechanism for scaling context length without prohibitive cost.

6.2 Links to recent works from 2024 to 2025

Although the constituent articles were completed between 2020 and 2023, the surrounding landscape has moved quickly. To situate these contributions in the present, this section offers a focused update on how closely related ideas evolved during 2024–2025, highlighting representative developments rather than attempting an exhaustive survey. The goal is to clarify where the proposed mechanisms have been extended or refined, where independent lines of work have converged on similar design choices, and where open problems remain.

Text extraction (retrieval and n-gram tables) for efficient conditional generation. A growing body of work argues that before generating, systems should choose what to show the model. For long inputs, one representative 2024 study *On Context Utilization in Summarization with LLMs* [240] shows pronounced position bias and introduces the MiddleSum benchmark, finding that hierarchical or incremental schemes that pre-structure the source mitigate the bias. This supports the thesis's idea of a compact sketch or extract before abstraction. In retrieval-augmented generation, *RankRAG* [241] is an influential system that aligns closely with this thesis's selection-then-generation pattern: a single instruction-tuned LLM performs context ranking and answer generation, effectively front-loading ranked content selection so that the generator conditions on higher-quality context. Efficiency trends also include n-gram/speculation-based acceleration that proposes candidate continuations cheaply and verifies them with the base model. For example, *N-Grammys* [242] and *SAM-Decoding* [243] show that using draft n-gram–driven lookups and finding the longest suffix match respectively can deliver notable speedups in autoregressive decoding.

Parameter-efficient multi-task adaptation. After well known adapters that came after our work such as *LoRA* [244], 2024 introduced *DoRA* [245], a weight-decomposed low-rank method that separates magnitude and direction to narrow the gap to full fine-tuning without inference overhead. DoRA's consistent gains across LLaMA/LLaVA/VL-BART suggest a practical path for stronger transfer under fixed update budgets. *Hyperdecoders* [246] an input-conditioned hypernetwork that produces instance-specific decoder adaptations (not just per-task), often outperforming PEFT and even full fine-tuning on several NLP tasks; good to cite if you want a stronger "conditioning signal".

Interactive disambiguation before decoding. A growing line of work now explicitly benchmarks and trains models to detect and clarify ambiguity before answering. The *CLAMBER* benchmark [247] shows that off-the-shelf LLMs still struggle to identify ambiguous queries and craft good clarifying questions, underscoring the value of brief, targeted interactions prior to generation as proposed in this thesis.

Long-range modeling with hybrid SSM–attention layers. There has been rapid movement toward architectures that combine efficient state-space updates for long-horizon propagation with attention for local, content-aware mixing-precisely the design space explored by Block-State Transformers. At scale, the *Zamba* [248] family interleaves *Mamba* [249] (SSM) and attention to reach large contexts with favorable throughput-quality trade-offs. *Jet-Nemotron* [250] applies post–architecture search to a pretrained model freezing MLPs and selectively keeping, dropping, or replacing attention blocks-to achieve large speedups while matching strong baselines. It pursues the same goal as Block-State–style hybridization in our work but instead of SSMs, Jet-Nemotron uses *Gated Delta Networks* [251], a Mamba-style state-space extension.

6.3 Future Research

A central bet of this dissertation is that the path to more capable AI runs through efficiency, not just scale. Biological intelligence offers a provocative benchmark: a human brain operates on roughly 20 W while supporting rich, continual cognition. By contrast, modern AI systems often require orders of magnitude more energy for far narrower tasks. Treating energy, memory footprint, and latency as first-class objectives (coequal with accuracy) can unlock models that are both more useful and more widely deployable. Empirically grounded comparisons [252] remind us why this matters. The brain metabolism scales under a tight power budget per neuron (≈20 W total), whereas mainstream GPU inference routinely draws hundreds of watts per device, and the environmental and economic costs of today's large models remain substantial.

Efficiency is not merely about cost; it enables qualitatively new behaviors. Compute-optimal training [253] (e.g., Chinchilla-style data/parameter balancing) improved downstream efficiency and accuracy simultaneously, showing that better use of tokens and parameters can beat brute-force scaling. On the inference side, test-time compute (TTC) has emerged as a controllable knob. We can allocate more "thinking time" can raise reasoning quality, but it must be budgeted and steered to remain practical[1]. Recent system cards and research notes highlight smooth gains from increased train and test time compute, yet also emphasize the need for principled scheduling and verification so that extra thinking yields reliable improvements rather than waste.

Building on this research and new methods and use-cases tested since 2023, I have started exploring:

1. Adaptive test-time compute (TTC) which allocates extra chains, searches, or verifications only when the model's own uncertainty, disagreement among samples, or verifier scores justify it. This aligns with recent TTC work that scales sampling and verification while keeping a tight budget. Low-entropy cases stay cheap and hard cases merit deeper search with verifiers or self-consistency.

2. Future work should treat fast drafting and parallel refinement as first-class. We can combine retrieval-aware speculation with verification, then explore highly parallel gen-

[1] See https://openai.com/index/learning-to-reason-with-llms

erators such as diffusion-style LLMs in the spirit of Mercury [254] for multi-token refinement, and continuous autoregressive schemes that compress K tokens into a single vector step as in recent CALM [255] designs. The research goal is a unified stack that learns when to draft, when to refine in parallel, and when to fall back to standard decoding, all under a fixed test-time compute budget. Success here would lower latency and energy per answer while preserving the target distribution and quality.

3. Architectures that move work from memory traffic to structured state. Biological efficiency stems partly from locality and sparse activation. Hybrid SSM–attention layers are a step in this direction. Future work will push stateful, streaming representations that minimize KV-cache bandwidth, reduce random memory access, and exploit structured recurrence. This direction also complements emerging systems work that treats inference scheduling and memory movement as core optimization problems with large energy dividends.

4. Co-design with hardware and runtime compression/optimization. Future work will make models runtime-aware without rehashing decoding strategies. We started to focus on: (i) budget-aware layers that minimize memory movement and KV-cache traffic; (ii) dynamic precision and calibration that adjust per layer/request under accuracy guards; (iii) on-the-fly compression of activations and caches (low-rank/entropy coding, segment pruning, input-aware compression) with learned admission policies; and (iv) autotuned scheduling that optimizes prefetch/eviction and device placement under power caps. The objective is a portable serving stack that jointly minimizes joules and latency per answer while preserving quality.

REFERENCES

[1] E. Sharma, C. Li, and L. Wang, “Bigpatent: A large-scale dataset for abstractive and coherent summarization,” *arXiv preprint arXiv:1906.03741*, 2019.

[2] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, “A discourse-aware attention model for abstractive summarization of long documents,” *CoRR*, vol. abs/1804.05685, 2018. [Online]. Available: http://arxiv.org/abs/1804.05685

[3] M. Grusky, M. Naaman, and Y. Artzi, “Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 708–719. [Online]. Available: https://aclanthology.org/N18-1065

[4] ——, “Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies,” *arXiv preprint arXiv:1804.11283*, 2018.

[5] A. Mendes, S. Narayan, S. Miranda, Z. Marinho, A. F. Martins, and S. B. Cohen, “Jointly extracting and compressing documents with summary state representations,” *arXiv preprint arXiv:1904.02020*, 2019.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[7] A. C. Stickland, I. Murray, someone, and someone, “BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning,” ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 5986–5995. [Online]. Available: http://proceedings.mlr.press/v97/stickland19a.html

[8] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-efficient transfer learning for NLP,” *CoRR*, vol. abs/1902.00751, 2019. [Online]. Available: http://arxiv.org/abs/1902.00751

[9] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," *CoRR*, vol. abs/1901.11504, 2019. [Online]. Available: http://arxiv.org/abs/1901.11504

[10] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao, "SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Online: Association for Computational Linguistics, Jul. 2020, pp. 2177–2190. [Online]. Available: https://aclanthology.org/2020.acl-main.197

[11] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing.* Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 388–395. [Online]. Available: https://aclanthology.org/W04-3250

[12] D. Hutchins, I. Schlag, Y. Wu, E. Dyer, and B. Neyshabur, "Block-recurrent transformers," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=uloenYmLCAo

[13] H. Mehta, A. Gupta, A. Cutkosky, and B. Neyshabur, "Long range language modeling via gated state spaces," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=5MkYIYCbva

[14] J. Phang, T. Févry, and S. R. Bowman, "Sentence encoders on STILTs: Supplementary training on intermediate labeled-data tasks," *CoRR*, vol. abs/1811.01088, 2018. [Online]. Available: http://arxiv.org/abs/1811.01088

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[16] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP.* Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. [Online]. Available: https://aclanthology.org/W18-5446

[17] A. Warstadt, A. Singh, and S. R. Bowman, "Neural network acceptability judgments," *CoRR*, vol. abs/1805.12471, 2018. [Online]. Available: http://arxiv.org/abs/1805.12471

[18] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.* Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: https://www.aclweb.org/anthology/D13-1170

[19] W. B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. [Online]. Available: https://www.aclweb.org/anthology/I05-5002

[20] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017).* Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14. [Online]. Available: https://www.aclweb.org/anthology/S17-2001

[21] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1112–1122. [Online]. Available: https://www.aclweb.org/anthology/N18-1101

[22] H. J. Levesque, "The winograd schema challenge." in *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning.* AAAI, 2011. [Online]. Available: http://dblp.uni-trier.de/db/conf/aaaiss/aaaiss2011-6.html#Levesque11

[23] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," *CoRR*, vol. abs/1905.00537, 2019. [Online]. Available: http://arxiv.org/abs/1905.00537

[24] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova, "BoolQ: Exploring the surprising difficulty of natural yes/no questions," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and*

Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 2924–2936. [Online]. Available: https://www.aclweb.org/anthology/N19-1300

[25] M.-C. de Marneffe, M. Simons, and J. Tonhauser, “The commitmentbank: Investigating projection in naturally occurring discourse,” *Proceedings of Sinn und Bedeutung*, vol. 23, no. 2, pp. 107–124, Jul. 2019. [Online]. Available: https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/601

[26] A. Gordon, Z. Kozareva, and M. Roemmele, “SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning,” in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012).* Montréal, Canada: Association for Computational Linguistics, 7-8 Jun. 2012, pp. 394–398. [Online]. Available: https://www.aclweb.org/anthology/S12-1052

[27] D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth, “Looking beyond the surface: A challenge set for reading comprehension over multiple sentences,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 252–262. [Online]. Available: https://www.aclweb.org/anthology/N18-1023

[28] S. Zhang, X. Liu, J. Liu, J. Gao, K. Duh, and B. V. Durme, “Record: Bridging the gap between human and machine commonsense reading comprehension,” *CoRR*, vol. abs/1810.12885, 2018. [Online]. Available: http://arxiv.org/abs/1810.12885

[29] A. Poliak, A. Haldar, R. Rudinger, J. E. Hu, E. Pavlick, A. S. White, and B. Van Durme, “Collecting diverse natural language inference problems for sentence representation evaluation,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 67–81. [Online]. Available: https://www.aclweb.org/anthology/D18-1007

[30] A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi, and D. Chen, “MRQA 2019 shared task: Evaluating generalization in reading comprehension,” in *Proceedings of the 2nd Workshop on Machine Reading for Question Answering.* Hong Kong, China:

Association for Computational Linguistics, Nov. 2019, pp. 1–13. [Online]. Available: https://www.aclweb.org/anthology/D19-5801

[31] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.* Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. [Online]. Available: https://www.aclweb.org/anthology/D16-1264

[32] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman, "NewsQA: A machine comprehension dataset," in *Proceedings of the 2nd Workshop on Representation Learning for NLP.* Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 191–200. [Online]. Available: https://www.aclweb.org/anthology/W17-2623

[33] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, "TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1601–1611. [Online]. Available: https://www.aclweb.org/anthology/P17-1147

[34] M. Dunn, L. Sagun, M. Higgins, V. U. Güney, V. Cirik, and K. Cho, "Searchqa: A new q&a dataset augmented with context from a search engine," *CoRR*, vol. abs/1704.05179, 2017. [Online]. Available: http://arxiv.org/abs/1704.05179

[35] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2369–2380. [Online]. Available: https://www.aclweb.org/anthology/D18-1259

[36] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: a benchmark for question answering research," *Transactions of the Association of Computational Linguistics*, 2019.

[37] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empiri-*

cal Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2015.

[38] T. Khot, A. Sabharwal, and P. Clark, “Scitail: A textual entailment dataset from science question answering,” in *AAAI*, 2018.

[39] J. Glover and C. Hokamp, “Task selection policies for multitask learning,” *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1907.06214

[40] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.* Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[41] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[42] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *Proceedings of Workshop at ICLR*, vol. 2013, 01 2013.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017.

[44] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, 2020.

[45] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal *et al.*, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, 2020.

[46] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of NAACL-HLT*, 2019.

[47] L. Ouyang, J. Wu, X. Jiang *et al.*, “Training language models to follow instructions with human feedback,” *arXiv preprint arXiv:2203.02155*, 2022.

[48] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *Proceedings of ICLR*, 2019.

[49] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of ACL*, 2021.

[50] J. Pilault, A. E. hattami, and C. Pal, "Conditionally adaptive multi-task learning: Improving transfer learning in {nlp} using fewer parameters & less data," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=de11dbHzAMF

[51] J. Pilault, C. Liu, M. Bansal, and M. Dreyer, "On conditional and compositional language model differentiable prompting," 2023. [Online]. Available: https://arxiv.org/abs/2307.01446

[52] E. Ben Zaken, Y. Goldberg, and S. Ravfogel, "BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1–9. [Online]. Available: https://aclanthology.org/2022.acl-short.1

[53] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Computing Surveys*, 2022.

[54] J. Pilault, R. Li, S. Subramanian, and C. Pal, "On extractive and abstractive neural document summarization with transformer language models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 9308–9319. [Online]. Available: https://aclanthology.org/2020.emnlp-main.748

[55] J. Pilault, M. Fathi, O. Firat, C. Pal, P.-L. Bacon, and R. Goroshin, "Block-state transformers," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 7311–7329. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/16ccd203e9e3696a7ab0dcf568316379-Paper-Conference.pdf

[56] J. Pilault, X. Garcia, A. Bražinskas, and O. Firat, "Interactive-chain-prompting: Ambiguity resolution for crosslingual conditional generation with interaction," in *Proceedings of the 13th International Joint Conference on Natural Language*

Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), J. C. Park, Y. Arase, B. Hu, W. Lu, D. Wijaya, A. Purwarianti, and A. A. Krisnadhi, Eds. Nusa Dua, Bali: Association for Computational Linguistics, Nov. 2023, pp. 455–483. [Online]. Available: https://aclanthology.org/2023.ijcnlp-main.31/

[57] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, "Adapterfusion: Non-destructive task composition for transfer learning," in *Proceedings of EACL*, 2021.

[58] E. Ben-Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of ACL*, 2022.

[59] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *Proceedings of ICLR*, 2021.

[60] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," in *Proceedings of ICLR*, 2021.

[61] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, "Mixed precision training," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1gs9JgRZ

[62] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 16 344–16 359. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/67d57c32e20fd0a7a302cb81d36e40d5-Paper-Conference.pdf

[63] M. Ott, S. Edunov, D. Grangier, and M. Auli, "Scaling neural machine translation," in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 1–9. [Online]. Available: https://aclanthology.org/W18-6301

[64] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, "Zero: Memory optimization towards training A trillion parameter models," *CoRR*, vol. abs/1910.02054, 2019. [Online]. Available: http://arxiv.org/abs/1910.02054

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[66] G. Stanovsky, N. A. Smith, and L. Zettlemoyer, "Evaluating gender bias in machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1679–1684. [Online]. Available: https://aclanthology.org/P19-1164

[67] S. L. Blodgett, S. Barocas, H. Daumé III, and H. Wallach, "Language (technology) is power: A critical survey of "bias" in NLP," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 5454–5476. [Online]. Available: https://aclanthology.org/2020.acl-main.485

[68] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. Daumé III, and K. Crawford, "Datasheets for datasets," *Communications of the ACM*, vol. 64, no. 12, pp. 86–92, December 2021. [Online]. Available: https://www.microsoft.com/en-us/research/publication/datasheets-for-datasets/

[69] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 2633–2650. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting

[70] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 27 730–27 744. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

[71] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon *et al.*, "Constitutional ai: Harmlessness from ai feedback," *arXiv preprint arXiv:2212.08073*, 2022.

[72] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[73] R. Nallapati, B. Zhou, and M. Ma, "Classify or select: Neural architectures for extractive document summarization," *arXiv preprint arXiv:1611.04244*, 2016.

[74] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," *arXiv preprint arXiv:1603.07252*, 2016.

[75] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," *arXiv preprint arXiv:1805.11080*, 2018.

[76] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," pp. 2692–2700, 2015.

[77] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *CoRR*, vol. abs/1509.00685, 2015. [Online]. Available: http://arxiv.org/abs/1509.00685

[78] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv preprint arXiv:1602.06023*, 2016.

[79] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 93–98.

[80] J. Gu, Z. Lu, H. Li, and V. O. K. Li, "Incorporating copying mechanism in sequence-to-sequence learning," *CoRR*, vol. abs/1603.06393, 2016. [Online]. Available: http://arxiv.org/abs/1603.06393

[81] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," *arXiv preprint arXiv:1603.08148*, 2016.

[82] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *CoRR*, vol. abs/1704.04368, 2017. [Online]. Available: http://arxiv.org/abs/1704.04368

[83] S. Gehrmann, Y. Deng, and A. M. Rush, "Bottom-up abstractive summarization," *arXiv preprint arXiv:1808.10792*, 2018.

[84] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, and M. Sun, "A unified model for extractive and abstractive summarization using inconsistency loss," *arXiv preprint arXiv:1805.06266*, 2018.

[85] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," *arXiv preprint arXiv:1801.10198*, 2018.

[86] R. Mihalcea and P. Tarau, “Textrank: Bringing order into text,” in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[87] R. Nallapati, F. Zhai, and B. Zhou, “Summarunner: A recurrent neural network based sequence model for extractive summarization of documents,” 2017.

[88] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[89] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” *arXiv preprint arXiv:1508.04025*, 2015.

[90] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[91] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh *et al.*, “Mixed precision training,” *arXiv preprint arXiv:1710.03740*, 2017.

[92] A. Fan, M. Lewis, and Y. Dauphin, “Hierarchical neural story generation,” *arXiv preprint arXiv:1805.04833*, 2018.

[93] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” *arXiv preprint arXiv:1508.07909*, 2015.

[94] C.-Y. Lin, “Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough?” in *NTCIR*, 2004.

[95] J.-P. Ng and V. Abrecht, “Better summarization evaluation with word embeddings for ROUGE,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1925–1930. [Online]. Available: https://aclanthology.org/D15-1222

[96] G. Erkan and D. R. Radev, “Lexrank: Graph-based lexical centrality as salience in text summarization,” *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[97] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, “Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion,” *Information Processing & Management*, vol. 43, no. 6, pp. 1606–1618, 2007.

[98] J. Steinberger and K. Jezek, “Using latent semantic analysis in text summarization and summary evaluation,” *Proc. ISIM*, vol. 4, pp. 93–100, 2004.

[99] R. Paulus, C. Xiong, and R. Socher, “A deep reinforced model for abstractive summarization,” *arXiv preprint arXiv:1705.04304*, 2017.

[100] Y. Graham, “Re-evaluating automatic summarization with bleu and 192 shades of rouge,” pp. 128–137, 2015.

[101] N. Weber, L. Shekhar, N. Balasubramanian, and K. Cho, “Controlling decoding for more abstractive summaries with copy-based networks,” *arXiv preprint arXiv:1803.07038*, 2018.

[102] K. Lee, O. Firat, A. Agarwal, C. Fannjiang, and D. Sussillo, “Hallucinations in neural machine translation,” 2019. [Online]. Available: https://openreview.net/forum?id=SkxJ-309FQ

[103] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, “Pegasus: Pre-training with extracted gap-sentences for abstractive summarization,” in *ICML*, 2020.

[104] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[105] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[106] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” pp. 2672–2680, 2014.

[107] M. E. Peters, M. Neumann, L. Zettlemoyer, and W. Yih, “Dissecting contextual word embeddings: Architecture and representation,” *CoRR*, vol. abs/1808.08949, 2018. [Online]. Available: http://arxiv.org/abs/1808.08949

[108] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith, “Linguistic knowledge and transferability of contextual representations,” *CoRR*, vol. abs/1903.08855, 2019. [Online]. Available: http://arxiv.org/abs/1903.08855

[109] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *ICML*, 2008, pp. 160–167. [Online]. Available: https://doi.org/10.1145/1390156.1390177

[110] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[111] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv*, pp. arXiv–2005, 2020.

[112] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does BERT look at? an analysis of BERT's attention," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286. [Online]. Available: https://www.aclweb.org/anthology/W19-4828

[113] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," *CoRR*, vol. abs/1905.05950, 2019. [Online]. Available: http://arxiv.org/abs/1905.05950

[114] A. Merchant, E. Rahimtoroghi, E. Pavlick, and I. Tenney, "What happens to bert embeddings during fine-tuning?" *arXiv preprint arXiv:2004.14448*, 2020.

[115] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[116] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 328–339. [Online]. Available: https://aclanthology.org/P18-1031

[117] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997. [Online]. Available: https://doi.org/10.1023/A:1007379606734

[118] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392.

[119] S. Ruder, “An overview of multi-task learning in deep neural networks,” *ArXiv*, vol. abs/1706.05098, 2017.

[120] R. Caruana, “Multitask learning: A knowledge-based source of inductive bias,” in *Proceedings of the Tenth International Conference on Machine Learning.* Morgan Kaufmann, 1993, pp. 41–48.

[121] Y. Zhang and Q. Yang, “A survey on multi-task learning,” *CoRR*, vol. abs/1707.08114, 2017. [Online]. Available: http://arxiv.org/abs/1707.08114

[122] A. Wang, J. Hula, P. Xia, R. Pappagari, R. T. McCoy, R. Patel, N. Kim, I. Tenney, Y. Huang, K. Yu, S. Jin, B. Chen, B. Van Durme, E. Grave, E. Pavlick, and S. R. Bowman, “Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[123] S. Wu, H. R. Zhang, and C. Ré, “Understanding and improving information transfer in multi-task learning,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SylzhkBtDB

[124] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, “Gradient surgery for multi-task learning,” *arXiv preprint arXiv:2001.06782*, 2020.

[125] J. Serrà, D. Suris, M. Miron, and A. Karatzoglou, “Overcoming catastrophic forgetting with hard attention to the task,” in *ICML*, 2018, pp. 4555–4564. [Online]. Available: http://proceedings.mlr.press/v80/serra18a.html

[126] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, “Film: Visual reasoning with a general conditioning layer,” in *AAAI*, 2018.

[127] H. de Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, “Modulating early visual processing by language,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 6594–6604. [Online]. Available: http://papers.nips.cc/paper/7237-modulating-early-visual-processing-by-language.pdf

[128] L. J. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *CoRR*, vol. abs/1607.06450, 2016. [Online]. Available: http://arxiv.org/abs/1607.06450

[129] E. Collins, N. Rozanov, and B. Zhang, "Evolutionary data measures: Understanding the difficulty of text classification tasks," in *Proceedings of the 22nd Conference on Computational Natural Language Learning.* Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 380–391. [Online]. Available: https://aclanthology.org/K18-1037

[130] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[131] M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei, "Dynamic task prioritization for multitask learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[132] J. Bingel and A. Søgaard, "Identifying beneficial task relations for multi-task learning in deep neural networks," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers.* Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 164–169. [Online]. Available: https://aclanthology.org/E17-2026

[133] E. Kerinec, C. Braud, and A. Søgaard, "When does deep multi-task learning work for loosely related document classification tasks?" in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP.* Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 1–8. [Online]. Available: https://aclanthology.org/W18-5401

[134] T. Standley, A. R. Zamir, D. Chen, L. J. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" *CoRR*, vol. abs/1905.07553, 2019. [Online]. Available: http://arxiv.org/abs/1905.07553

[135] Y. Pruksachatkun, J. Phang, H. Liu, P. M. Htut, X. Zhang, R. Y. Pang, C. Vania, K. Kann, and S. R. Bowman, "Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work?" *arXiv preprint arXiv:2005.00628*, 2020.

[136] K. Clark, M. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le, "Bam! born-again multi-task networks for natural language understanding," *CoRR*, vol. abs/1907.04829, 2019. [Online]. Available: http://arxiv.org/abs/1907.04829

[137] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, “The natural language decathlon: Multitask learning as question answering,” *arXiv preprint arXiv:1806.08730*, 2018.

[138] R. Aharoni, M. Johnson, and O. Firat, “Massively multilingual neural machine translation,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 3874–3884. [Online]. Available: https://aclanthology.org/N19-1388

[139] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. V. Durme, S. R. Bowman, D. Das, and E. Pavlick, “What do you learn from context? probing for sentence structure in contextualized word representations,” *CoRR*, vol. abs/1905.06316, 2019. [Online]. Available: http://arxiv.org/abs/1905.06316

[140] Y. Tay, Z. Zhao, D. Bahri, D. Metzler, and D.-C. Juan, “Hypergrid: Efficient multi-task transformers with grid-wise decomposable hyper projections,” *arXiv preprint arXiv:2007.05891*, 2020.

[141] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” 2019.

[142] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, “Gshard: Scaling giant models with conditional computation and automatic sharding,” *arXiv preprint arXiv:2006.16668*, 2020.

[143] H. Peng, R. Schwartz, D. Li, and N. A. Smith, “A mixture of h - 1 heads is better than h heads,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 6566–6577. [Online]. Available: https://aclanthology.org/2020.acl-main.587

[144] O. Sener and V. Koltun, “Multi-task learning as multi-objective optimization,” *CoRR*, vol. abs/1810.04650, 2018. [Online]. Available: http://arxiv.org/abs/1810.04650

[145] Z. Chen, V. Badrinarayanan, C. Lee, and A. Rabinovich, “Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks,” *CoRR*, vol. abs/1711.02257, 2017. [Online]. Available: http://arxiv.org/abs/1711.02257

[146] F. Ikhwantri, S. Louvan, K. Kurniawan, B. Abisena, V. Rachman, A. F. Wicaksono, and R. Mahendra, “Multi-task active learning for neural semantic role labeling on

low resource conversational corpus," in *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, 2018, pp. 43–50.

[147] R. Reichart, K. Tomanek, U. Hahn, and A. Rappoport, "Multi-task active learning for linguistic annotations," in *Proceedings of ACL-08: HLT*, 2008, pp. 861–869.

[148] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *CoRR*, vol. abs/1705.07115, 2017. [Online]. Available: http://arxiv.org/abs/1705.07115

[149] W. Chen, Y. Zhang, and H. Isahara, "An empirical study of Chinese chunking," in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Sydney, Australia: Association for Computational Linguistics, Jul. 2006, pp. 97–104. [Online]. Available: https://aclanthology.org/P06-2013

[150] D. Charles, M. Chickering, and P. Simard, "Counterfactual reasoning and learning systems: The example of computational advertising," *Journal of Machine Learning Research*, vol. 14, pp. 3207–3260, November 2013.

[151] J. von Oswald, C. Henning, B. F. Grewe, and J. Sacramento, "Continual learning with hypernetworks," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SJgwNerKvB

[152] E. M. Ponti, I. Vulić, R. Cotterell, M. Parovic, R. Reichart, and A. Korhonen, "Parameter space factorization for zero-shot learning across tasks and languages," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 410–428, 2021. [Online]. Available: https://aclanthology.org/2021.tacl-1.25

[153] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, no. 4, pp. 1303–1347, 2013. [Online]. Available: http://jmlr.org/papers/v14/hoffman13a.html

[154] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *CoRR*, vol. abs/1910.03771, 2019. [Online]. Available: http://arxiv.org/abs/1910.03771

[155] P. He, X. Liu, W. Chen, and J. Gao, "A hybrid neural network model for commonsense reasoning," in *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 13–21. [Online]. Available: https://aclanthology.org/D19-6002

[156] L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham, “Results of the WNUT2017 shared task on novel and emerging entity recognition,” in *Proceedings of the 3rd Workshop on Noisy User-generated Text.* Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 140–147. [Online]. Available: https://aclanthology.org/W17-4418

[157] D. Q. Nguyen, T. Vu, and A. T. Nguyen, “Bertweet: A pre-trained language model for english tweets,” *arXiv preprint arXiv:2005.10200*, 2020.

[158] X. Liu, H. Cheng, P. He, W. Chen, Y. Wang, H. Poon, and J. Gao, “Adversarial training for large neural language models,” 2020.

[159] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. [Online]. Available: https://aclanthology.org/2020.acl-main.747

[160] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[161] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[162] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato,

R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[163] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *CoRR*, vol. abs/2001.08361, 2020. [Online]. Available: https://arxiv.org/abs/2001.08361

[164] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, C. Hallacy, B. Mann, A. Radford, A. Ramesh, N. Ryder, D. M. Ziegler, J. Schulman, D. Amodei, and S. McCandlish, "Scaling laws for autoregressive generative modeling," *CoRR*, vol. abs/2010.14701, 2020. [Online]. Available: https://arxiv.org/abs/2010.14701

[165] D. Hernandez, J. Kaplan, T. Henighan, and S. McCandlish, "Scaling laws for transfer," *CoRR*, vol. abs/2102.01293, 2021. [Online]. Available: https://arxiv.org/abs/2102.01293

[166] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "{GS}hard: Scaling giant models with conditional computation and automatic sharding," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=qrwe7XHTmYb

[167] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022, survey Certification. [Online]. Available: https://openreview.net/forum?id=yzkSU5zdwD

[168] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, H. F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, E. Rutherford, T. Hennigan, J. Menick, A. Cassirer, R. Powell, G. van den Driessche, L. A. Hendricks, M. Rauh, P. Huang, A. Glaese, J. Welbl, S. Dathathri, S. Huang, J. Uesato, J. Mellor, I. Higgins, A. Creswell, N. McAleese, A. Wu, E. Elsen, S. M. Jayakumar, E. Buchatskaya, D. Budden, E. Sutherland, K. Simonyan, M. Paganini, L. Sifre, L. Martens, X. L. Li, A. Kuncoro, A. Nematzadeh, E. Gribovskaya, D. Donato, A. Lazaridou, A. Mensch, J. Lespiau, M. Tsimpoukelli, N. Grigorev, D. Fritz, T. Sottiaux, M. Pajarskas, T. Pohlen, Z. Gong, D. Toyama, C. de Masson d'Autume, Y. Li, T. Terzi, V. Mikulik, I. Babuschkin, A. Clark, D. de Las Casas, A. Guy, C. Jones, J. Bradbury, M. Johnson,

B. A. Hechtman, L. Weidinger, I. Gabriel, W. S. Isaac, E. Lockhart, S. Osindero, L. Rimell, C. Dyer, O. Vinyals, K. Ayoub, J. Stanway, L. Bennett, D. Hassabis, K. Kavukcuoglu, and G. Irving, "Scaling language models: Methods, analysis & insights from training gopher," *CoRR*, vol. abs/2112.11446, 2021. [Online]. Available: https://arxiv.org/abs/2112.11446

[169] B. Ghorbani, O. Firat, M. Freitag, A. Bapna, M. Krikun, X. Garcia, C. Chelba, and C. Cherry, "Scaling laws for neural machine translation," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=hR_SMu8cxCV

[170] J. Wei, X. Wang, D. Schuurmans, M. Bosma, brian ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=_VjQlMeSB_J

[171] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le, and E. Chi, "Least-to-most prompting enables complex reasoning in large language models," 2022. [Online]. Available: https://arxiv.org/abs/2205.10625

[172] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "Palm: Scaling language modeling with pathways," 2022. [Online]. Available: https://arxiv.org/abs/2204.02311

[173] D. Vilar, M. Freitag, C. Cherry, J. Luo, V. Ratnakar, and G. Foster, "Prompting palm for translation: Assessing strategies and performance," 2022. [Online]. Available: https://arxiv.org/abs/2211.09102

[174] B. Zhang, B. Haddow, and A. Birch, "Prompting large language model for machine translation: A case study," 2023. [Online]. Available: https://arxiv.org/abs/2301.07069

[175] T. Wu, M. Terry, and C. J. Cai, “Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts,” in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3491102.3517582

[176] T. Badeka, “Machine translation: Search queries at ebay,” 2016. [Online]. Available: https://tech.ebayinc.com/engineering/machine-translation-search-queries-at-ebay/

[177] P. Lison and J. Tiedemann, “OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16).* Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 923–929. [Online]. Available: https://aclanthology.org/L16-1147

[178] X. Garcia and O. Firat, “Using natural language prompts for machine translation,” 2022. [Online]. Available: https://arxiv.org/abs/2202.11822

[179] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, “mT5: A massively multilingual pre-trained text-to-text transformer,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Online: Association for Computational Linguistics, Jun. 2021, pp. 483–498. [Online]. Available: https://aclanthology.org/2021.naacl-main.41

[180] K. Lee, M.-W. Chang, and K. Toutanova, “Latent retrieval for weakly supervised open domain question answering,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 6086–6096. [Online]. Available: https://aclanthology.org/P19-1612

[181] M. Aliannejadi, H. Zamani, F. Crestani, and W. B. Croft, “Asking clarifying questions in open-domain information-seeking conversations,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 475–484. [Online]. Available: https://doi.org/10.1145/3331184.3331265

[182] H. Zamani, G. Lueck, E. Chen, R. Quispe, F. Luu, and N. Craswell, “MIMICS: A large-scale data collection for search clarification,” *CoRR*, vol. abs/2006.10174, 2020. [Online]. Available: https://arxiv.org/abs/2006.10174

[183] K. D. Dhole, “Resolving intent ambiguities by retrieving discriminative clarifying questions,” *ArXiv*, vol. abs/2008.07559, 2020.

[184] J. Wang and W. Li, “Template-guided clarifying question generation for web search clarification,” in *Proceedings of the 30th ACM International Conference on Information; Knowledge Management*, ser. CIKM ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 3468–3472. [Online]. Available: https://doi.org/10.1145/3459637.3482199

[185] Z. Wu, R. Parish, H. Cheng, S. Min, P. Ammanabrolu, M. Ostendorf, and H. Hajishirzi, “Inscit: Information-seeking conversations with mixed-initiative interactions,” 2022. [Online]. Available: https://arxiv.org/abs/2207.00746

[186] D. Krasheninnikov, E. Krasheninnikov, and D. Krueger, “Assistance with large language models,” in *NeurIPS ML Safety Workshop*, 2022. [Online]. Available: https://openreview.net/forum?id=OE9V81spp6B

[187] S. Min, J. Michael, H. Hajishirzi, and L. Zettlemoyer, “AmbigQA: Answering ambiguous open-domain questions,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 5783–5797. [Online]. Available: https://aclanthology.org/2020.emnlp-main.466

[188] M. Ware, E. FRANK, G. HOLMES, M. HALL, and I. H. WITTEN, “Interactive machine learning: letting users build classifiers,” *International Journal of Human-Computer Studies*, vol. 55, no. 3, pp. 281–292, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1071581901904999

[189] J. A. Fails and D. R. Olsen, “Interactive machine learning,” in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, ser. IUI ’03. New York, NY, USA: Association for Computing Machinery, 2003, p. 39–45. [Online]. Available: https://doi.org/10.1145/604045.604056

[190] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, “Power to the people: The role of humans in interactive machine learning,” *AI Magazine*, vol. 35, no. 4, pp. 105–120, Dec. 2014. [Online]. Available: https://ojs.aaai.org/index.php/aimagazine/article/view/2513

[191] S. Green, J. Chuang, J. Heer, and C. D. Manning, “Predictive translation memory: A mixed-initiative system for human language translation,” in

ACM User Interface Software & Technology (UIST), 2014. [Online]. Available: http://idl.cs.washington.edu/papers/ptm

[192] S. Santy, S. Dandapat, M. Choudhury, and K. Bali, "INMT: Interactive neural machine translation prediction," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 103–108. [Online]. Available: https://aclanthology.org/D19-3018

[193] N. Konstantinova and C. Orasan, *Interactive Question Answering*, 10 2013, pp. 149 –.

[194] J. Gao, M. Galley, and L. Li, "Neural approaches to conversational AI," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2–7. [Online]. Available: https://aclanthology.org/P18-5002

[195] S. Hussain, O. A. Sianaki, and N. Ababneh, "A survey on conversational agents/chatbots classification and design techniques," in *AINA Workshops*, 2019.

[196] T. Sellam, D. Das, and A. Parikh, "BLEURT: Learning robust metrics for text generation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7881–7892. [Online]. Available: https://aclanthology.org/2020.acl-main.704

[197] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman, "Nuanced metrics for measuring unintended bias with real data for text classification," in *Companion Proceedings of The 2019 World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 491–500. [Online]. Available: https://doi.org/10.1145/3308560.3317593

[198] D. Saunders and B. Byrne, "Reducing gender bias in neural machine translation as a domain adaptation problem," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7724–7736. [Online]. Available: https://aclanthology.org/2020.acl-main.690

[199] A. Stafanovičs, T. Bergmanis, and M. Pinnis, "Mitigating gender bias in machine translation with target gender annotations," in *Proceedings of the Fifth Conference on*

Machine Translation. Online: Association for Computational Linguistics, Nov. 2020, pp. 629–638. [Online]. Available: https://aclanthology.org/2020.wmt-1.73

[200] J. Wang, B. Rubinstein, and T. Cohn, "Measuring and mitigating name biases in neural machine translation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2576–2590. [Online]. Available: https://aclanthology.org/2022.acl-long.184

[201] E. Rippeth, S. Agrawal, and M. Carpuat, "Controlling translation formality using pre-trained multilingual language models," in *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022).* Dublin, Ireland (in-person and online): Association for Computational Linguistics, May 2022, pp. 327–340. [Online]. Available: https://aclanthology.org/2022.iwslt-1.30

[202] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[203] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107–116, 1998. [Online]. Available: http://dblp.uni-trier.de/db/journals/ijufks/ijufks6.html#Hochreiter98

[204] U. Khandelwal, H. He, P. Qi, and D. Jurafsky, "Sharp nearby, fuzzy far away: How neural language models use context," *CoRR*, vol. abs/1805.04623, 2018. [Online]. Available: http://arxiv.org/abs/1805.04623

[205] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[206] R. Thoppilan, D. D. Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, Y. Li, H. Lee, H. S. Zheng, A. Ghafouri, M. Menegali, Y. Huang, M. Krikun, D. Lepikhin, J. Qin, D. Chen, Y. Xu, Z. Chen, A. Roberts, M. Bosma, Y. Zhou, C. Chang, I. Krivokon, W. Rusch, M. Pickett, K. S. Meier-Hellstern, M. R. Morris, T. Doshi, R. D. Santos, T. Duke, J. Soraker, B. Zevenbergen, V. Prabhakaran, M. Diaz, B. Hutchinson, K. Olson, A. Molina, E. Hoffman-John,

J. Lee, L. Aroyo, R. Rajakumar, A. Butryna, M. Lamm, V. Kuzmina, J. Fenton, A. Cohen, R. Bernstein, R. Kurzweil, B. Aguera-Arcas, C. Cui, M. Croak, E. H. Chi, and Q. Le, “Lamda: Language models for dialog applications,” *CoRR*, vol. abs/2201.08239, 2022. [Online]. Available: https://arxiv.org/abs/2201.08239

[207] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, “Palm: Scaling language modeling with pathways,” 2022.

[208] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” *arXiv preprint arXiv:1901.02860*, 2019.

[209] R. Child, S. Gray, A. Radford, and I. Sutskever, “Generating long sequences with sparse transformers,” *CoRR*, vol. abs/1904.10509, 2019. [Online]. Available: http://arxiv.org/abs/1904.10509

[210] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang *et al.*, “Big bird: Transformers for longer sequences,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[211] Y. Tay, D. Bahri, L. Yang, D. Metzler, and D.-C. Juan, “Sparse Sinkhorn attention,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 9438–9447. [Online]. Available: https://proceedings.mlr.press/v119/tay20a.html

[212] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, “Linformer: Self-attention with linear complexity,” *CoRR*, vol. abs/2006.04768, 2020. [Online]. Available: https://arxiv.org/abs/2006.04768

[213] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlós, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. J. Colwell, and

A. Weller, “Rethinking attention with performers,” *CoRR*, vol. abs/2009.14794, 2020. [Online]. Available: https://arxiv.org/abs/2009.14794

[214] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, “Transformers are RNNs: Fast autoregressive transformers with linear attention,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 5156–5165. [Online]. Available: https://proceedings.mlr.press/v119/katharopoulos20a.html

[215] Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, and D. Metzler, “Long range arena: A benchmark for efficient transformers,” 2020.

[216] C. Li, M. Zhang, and Y. He, “The stability-efficiency dilemma: Investigating sequence length warmup for training GPT models,” in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=JpZ5du_Kdh

[217] S. Subramanian, R. Collobert, M. Ranzato, and Y. Boureau, “Multi-scale transformer language models,” *CoRR*, vol. abs/2005.00581, 2020. [Online]. Available: https://arxiv.org/abs/2005.00581

[218] A. Gu, K. Goel, and C. Ré, “Efficiently modeling long sequences with structured state spaces,” 2022.

[219] J. Cooley and J. Tukey, “An algorithm for the machine calculation of complex fourier series,” *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[220] T. Chihara, *An Introduction to Orthogonal Polynomials*, ser. Dover Books on Mathematics. Dover Publications, 2011. [Online]. Available: https://books.google.ca/books?id=71CVAwAAQBAJ

[221] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Re, “Hippo: Recurrent memory with optimal polynomial projections,” 2020.

[222] D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Ré, “Hungry hungry hippos: Towards language modeling with state space models,” 2023.

[223] Y. Wu, M. N. Rabe, D. Hutchins, and C. Szegedy, “Memorizing transformers,” in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=TrjbxzRcnf-

[224] X. Ma, C. Zhou, X. Kong, J. He, L. Gui, G. Neubig, J. May, and L. Zettlemoyer, “Mega: Moving average equipped gated attention,” 2023.

[225] M. Poli, S. Massaroli, E. Nguyen, D. Y. Fu, T. Dao, S. Baccus, Y. Bengio, S. Ermon, and C. Ré, “Hyena hierarchy: Towards larger convolutional language models,” 2023.

[226] D. Y. Fu, E. L. Epstein, E. Nguyen, A. W. Thomas, M. Zhang, T. Dao, A. Rudra, and C. Ré, “Simple hardware-efficient long convolutions for sequence modeling,” 2023.

[227] W. Hua, Z. Dai, H. Liu, and Q. Le, “Transformer quality in linear time,” in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 9099–9117. [Online]. Available: https://proceedings.mlr.press/v162/hua22a.html

[228] OpenAI, “Gpt-4 technical report,” 2023.

[229] J. T. H. Smith, A. Warrington, and S. W. Linderman, “Simplified state space layers for sequence modeling,” 2023.

[230] A. Gu, A. Gupta, K. Goel, and C. Ré, “On the parameterization and initialization of diagonal state space models,” 2022.

[231] A. Gupta, A. Gu, and J. Berant, “Diagonal state spaces are as effective as structured state spaces,” 2022.

[232] J. W. Rae, A. Potapenko, S. M. Jayakumar, C. Hillier, and T. P. Lillicrap, “Compressive transformers for long-range sequence modelling,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=SylKikSYDH

[233] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” 2019.

[234] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: composable transformations of Python+NumPy programs,” 2018. [Online]. Available: http://github.com/google/jax

[235] J. Heek, A. Levskaya, A. Oliver, M. Ritter, B. Rondepierre, A. Steiner, and M. van Zee, “Flax: A neural network library and ecosystem for JAX,” 2023. [Online]. Available: http://github.com/google/flax

[236] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization.” in *ICLR (Poster)*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14

[237] Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, and D. Metzler, “Long range arena : A benchmark for efficient transformers,” in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=qVyeW-grC2k

[238] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

[239] A. Gu, K. Goel, A. Gupta, and C. Ré, “On the parameterization and initialization of diagonal state space models,” in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=yJE7iQSAep

[240] M. Ravaut, A. Sun, N. Chen, and S. Joty, “On context utilization in summarization with large language models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 2764–2781. [Online]. Available: https://aclanthology.org/2024.acl-long.153/

[241] Y. Yu, W. Ping, Z. Liu, B. Wang, J. You, C. Zhang, M. Shoeybi, and B. Catanzaro, “Rankrag: Unifying context ranking with retrieval-augmented generation in llms,” in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 121 156–121 184. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/db93ccb6cf392f352570dd5af0a223d3-Paper-Conference.pdf

[242] L. Stewart, M. Trager, S. Gonugondla, and S. Soatto, “The N-Grammys: Accelerating autoregressive inference with learning-free batched speculation,” in *Proceedings of The 4th NeurIPS Efficient Natural Language and Speech Processing Workshop*, ser. Proceedings of Machine Learning Research, M. Rezagholizadeh,

P. Passban, S. Samiee, V. Partovi Nia, Y. Cheng, Y. Deng, Q. Liu, and B. Chen, Eds., vol. 262. PMLR, 14 Dec 2024, pp. 322–335. [Online]. Available: https://proceedings.mlr.press/v262/stewart24a.html

[243] Y. Hu, K. Wang, X. Zhang, F. Zhang, C. Li, H. Chen, and J. Zhang, "SAM decoding: Speculative decoding via suffix automaton," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 12 187–12 204. [Online]. Available: https://aclanthology.org/2025.acl-long.595/

[244] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[245] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen, "DoRA: Weight-decomposed low-rank adaptation," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR, 21–27 Jul 2024, pp. 32 100–32 121. [Online]. Available: https://proceedings.mlr.press/v235/liu24bn.html

[246] H. Ivison and M. Peters, "Hyperdecoders: Instance-specific decoders for multi-task NLP," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1715–1730. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.124/

[247] T. Zhang, P. Qin, Y. Deng, C. Huang, W. Lei, J. Liu, D. Jin, H. Liang, and T.-S. Chua, "CLAMBER: A benchmark of identifying and clarifying ambiguous information needs in large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 10 746–10 766. [Online]. Available: https://aclanthology.org/2024.acl-long.578/

[248] P. Glorioso, Q. Anthony, Y. Tokpanov, J. Whittington, J. Pilault, A. Ibrahim, and B. Millidge, “Zamba: A compact 7b ssm hybrid model,” *arXiv preprint arXiv:2405.16712*, 2024.

[249] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” in *First conference on language modeling*, 2024.

[250] Y. Gu, Q. Hu, H. Xi, J. Chen, S. Yang, S. Han, and H. Cai, “Jet-nemotron: Efficient language model with post neural architecture search,” in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=WZQXaTNYEB

[251] S. Yang, J. Kautz, and A. Hatamizadeh, “Gated delta networks: Improving mamba2 with delta rule,” in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=r8H7xhYPwz

[252] S. Herculano-Houzel, “Scaling of brain metabolism with a fixed energy budget per neuron: Implications for neuronal activity, plasticity and evolution,” *PLOS ONE*, vol. 6, no. 3, pp. 1–9, 03 2011. [Online]. Available: https://doi.org/10.1371/journal.pone.0017514

[253] X. Cheng, B. Chen, P. Li, J. Gong, J. Tang, and L. Song, “Training compute-optimal protein language models,” in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 69 386–69 418. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/8066ae1446b2bbccb5159587cc3b3bcc-Paper-Conference.pdf

[254] I. Labs, S. Khanna, S. Kharbanda, S. Li, H. Varma, E. Wang, S. Birnbaum, Z. Luo, Y. Miraoui, A. Palrecha, S. Ermon, A. Grover, and V. Kuleshov, “Mercury: Ultra-fast language models based on diffusion,” 2025. [Online]. Available: https://arxiv.org/abs/2506.17298

[255] C. Shao, D. Li, F. Meng, and J. Zhou, “Continuous autoregressive language models,” 2025. [Online]. Available: https://arxiv.org/abs/2510.27688

[256] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2015.

[257] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “Spanbert: Improving pre-training by representing and predicting spans,” *CoRR*, vol. abs/1907.10529, 2019. [Online]. Available: http://arxiv.org/abs/1907.10529

[258] M. Müller, A. Rios, E. Voita, and R. Sennrich, “A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation,” in *Proceedings of the Third Conference on Machine Translation: Research Papers.* Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 61–72. [Online]. Available: https://aclanthology.org/W18-6307

[259] R. Bawden, R. Sennrich, A. Birch, and B. Haddow, “Evaluating discourse phenomena in neural machine translation,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1304–1313. [Online]. Available: https://aclanthology.org/N18-1118

[260] E. Voita, R. Sennrich, and I. Titov, “When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1198–1212. [Online]. Available: https://aclanthology.org/P19-1116

[261] G. A. Miller, “WordNet: A lexical database for English,” in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. [Online]. Available: https://aclanthology.org/H94-1111

[262] S. Bird and E. Loper, “NLTK: The natural language toolkit,” in *Proceedings of the ACL Interactive Poster and Demonstration Sessions.* Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 214–217. [Online]. Available: https://aclanthology.org/P04-3031

[263] Y. Ogihara, “Baby names in japan, 2004–2018: common writings and their readings,” *BMC Research Notes*, vol. 13, no. 1, Dec. 2020, publisher Copyright: © 2020, The Author(s).

APPENDIX A LARGE TEXT SUMMARIZATION

A.1 Appendix: Large Text Summarization

A.1.1 Samples from the arXiv test set

Samples from the TLM-I and TLM-I+E (G,M) models on a random example from the arXiv test set.

A.1.2 Extractive Model Details

The model uses word embeddings of size 300. The token-level LSTM (sentence encoder), sentence-level LSTM (document encoder) and decoder each have 2 layers of 512 units and a dropout of 0.5 is applied at the output of each intermediate layer. We trained it with Adam, a learning rate 0.001, a weight decay of 10^{-5}, and using batch sizes of 32. We evaluate the model every 200 updates, using a patience of 50. At inference, we decode using beam search with a beam size of 4 for the pointer model and pick the k most likely sentences from the sentence classifier, where k is the average number of sentences in the summary across the training dataset.

TLM conditioned on intro — deep convolutional neural networks (cnns) have created new benchmarks in the object recognition challenge . most of these benchmark datasets contain thousands of images of interest . in addition to these datasets , several standard data - sets , such as pascal voc 2012 and the ms coco , have been used in conjunction with the same network in order to improve the object detection task . in this paper , we present a method for object localization using the deep convolutional neural networks trained for the simpler task of image - level classification . we propose localization candidates (or bounding boxes) which maximize classification scores for a class of interest . we explore the beam search strategy where the number of candidate bounding boxes are restricted as we traverse from the root node towards the leaf nodes . we then gradually explore smaller and smaller regions of interest till a point is reached where the classifier is not able to discriminate among the classes of interest . the detection candidates are organized in a search tree , the root node being represented by the entire test image . we then gradually explore smaller and smaller regions of interest till a point is reached where the classifier is not able to distinguish among the classes of interest .
TLM conditioned on intro + pointer network extracted sentences — deep convolutional neural networks (cnns) are widely used in image recognition applications . however , they face a number of challenges . first , cnn models are computationally expensive to train and to perform . second , they are not easy to implement and can not exploit global features . to alleviate these issues , we present a method that leverages the correlation between semantic information present in feature maps and localization of an object of interest . the method leverages feature map activations for object localization . we demonstrate that this method is significantly faster than state - of - the - art in localization accuracy by a significant margin of up to 8 map on two standard data - sets with complex scenes , pascal voc 2012 and the much larger ms coco .
Ground truth abstract — object localization is an important computer vision problem with a variety of applications . the lack of large scale object - level annotations and the relative abundance of image - level labels makes a compelling case for weak supervision in the object localization task . deep convolutional neural networks are a class of state-of-the-art methods for the related problem of object recognition . in this paper , we describe a novel object localization algorithm which uses classification networks trained on only image labels . this weakly supervised method leverages local spatial and semantic patterns captured in the convolutional layers of classification networks . we propose an efficient beam search based approach to detect and localize multiple objects in images . the proposed method significantly outperforms the state-of-the-art in standard object localization data - sets with a 8 point increase in map scores .

Table A.1 Qualitative Results - Generated abstracts from our models on a random example from the test set of [2]

APPENDIX B CONDITIONAL ADAPTIVE MULTITASK LEARNING

B.1 Appendix: Conditional Adaptive Multitask Learning

B.1.1 Summary of Acronyms

Acronyms of datasets and descriptions can be found below in section B.1.6.

Table B.1 List of acronyms used in this paper.

Acronym	Description
ARLM	Autoregressive Language Models
CA-MTL	Conditional Adaptive Multi-Task Learning: our architecture
CFF	Conditional Feed-Forward: a feed-forward layer modulated by a conditioning vector
CLN	Conditional Layer Normalization in section 3.2.1
EDM	Evolutionary Data Measures [129]: a task difficulty estimate
GLUE	General Language Understanding Evaluation [16]: a benchmark with multiple datasets
QA	Question Answering
MT	Multi-Task
MTAL	Multi-Task Active Learning: finding the most informative instance for multiple learners (or models)
MLM	Masked Language Model: BERT [6] is an example of an MLM
MTL	Multi-Task Learning: "learning tasks in parallel while using a shared representation" [117]
MRQA	Machine Reading for Question Answering [30]: a benchmark with multiple datasets
NER	Named Entity Recognition
NLP	Natural Language Processing
SOTA	State of the art
ST	Single Task fine-tuning: all weights are typically updated
ST-A	ST with Adapter modules: one adapter per task is trained and pretrained weights are optionally updated

B.1.2 Zero-Shot Results on SciTail and SNLI

Before testing models on domain adaptation in section 3.4.4, we ran zero-shot evaluations on the development set of SciTail and SNLI. Table B.2 outlines 8-task CA-MTL$_{\text{BERT-BASE}}$'s zero-shot transfer abilities when pretrained on GLUE with our MTL approach. We expand the task embedding layer to accommodate an extra task and explore various embedding initialization. We found that reusing STS-B and MRPC task embeddings worked best for SciTail and SNLI respectively.

B.1.3 More Experimental Details

We used a batch size of 32 and a seed of 12 in all experiments. We used Adam [256] as the optimizer with a learning rate of 2e-5. We applied a learning rate decay with warm up over the first 10% of the training steps. Unless otherwise specified, we used 5 epochs, a seed of 12 and a sequence length of 128. Additional details are outlined in section . Our data prepossessing and linear decoder heads are the same as in [6]. We used the same dropout

Table B.2 CA-MTL is flexible and extensible to new tasks. However, CA-MTL is sensitive to the new task's embedding. We tested multiple task embeddings that worked best on either SciTail or SNLI by checking performance in a zero shot setting or using 0% of the data.

Initialization of new task embedding layer	SciTail 0% of data	SNLI 0% of data
CoLA's embeddings	43.0	34.0
MNLI's embeddings	24.2	33.0
MRPC's embeddings	34.5	**45.5**
STS-B's embeddings	**46.9**	33.2
SST-2's embeddings	25.8	34.2
QQP's embeddings	31.7	37.3
QNLI's embeddings	32.0	38.0
RTE's embeddings	32.3	40.6
WNLI's embeddings	29.0	30.4
Average	28.7	37.7
Random initialization	46.8	34.0
Xavier initialization	29.8	37.6

rate of 0.1 in all layers. To run our experiments, we used either four NVIDIA P100 GPU for base models or four NVIDIA V100 GPU for larger ones. We did not perform parameter search. We do not use ensemble of models or task-specific tricks [6, 9, 136]. All models are either 12 Transformer layers for BASE and 24 Transformer layers for LARGE. Apart from CA-MTL, models trained in multi-task learning (BERT or RoBERTa without adapters) used random task sampling. For Table 3.1 and Figure 3.8, all BERT-based model have half their layers frozen (untrained) for a fair comparison of ablation results. For the 24-task MTL and CA-MTL models in Tables 3.4 and 3.5, we increased the input sequence length to 256 and used 8 epochs.

B.1.4 Baselines and Other Experimental Results

In this section, we present our baseline results for BERT, RoBERTa, CA-MTL as well as other models. Our single task results (ST) that we ran ourselves surpass other paper's reported scores in Table B.3. [15] reports random seed median scores for RoBERTa. However, our RoBERTa ST baseline **matches or surpasses** the original paper's scores **4 out 7 times** on the development set when scores are comparable (QQP F1 and STS-B spearman are not reported).

Table B.3 F1 scores are reported for QQP/MRPC, Spearman's correlation for STS-B, accuracy on the matched/mismatch sets for MNLI, Matthew's correlation for CoLA and accuracy for other tasks. ST=Single Task, MTL=Multitask. *QNLI v1 (we report v2) **F1 score or Spearman's correlation is not reported. ***Unknown random seeds. Results from: [1][7] [2][9] [3][14] [4][15].

Method	Total	Trained	GLUE								
	params	params/task	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg
Base Models — Dev set Results											
PALs+Anneal Samp.[1]	1.13×	12.5%	–	–	–	–	–	–	–	–	81.70
8-task CA-MTL$_{\text{BERT-BASE}}$	1.12×	5.6%	60.9	82.7/83.1	88.9	90.7	90.3	79.1	91.9	88.8	84.03
BERT LARGE Models — Dev set Results											
ST BERT-LARGE[2]	9×	100%	60.5	86.7/85.9	89.3	92.7*	89.3	70.1	94.9	86.5	84.0
ST BERT-LARGE[3]	9×	100%	62.1	86.2/86.2	92.3	89.4	88.5	70.0	92.5	90.1	84.1
ST BERT-LARGE	9×	100%	63.6	86.5/86.0	91.4	91.0	88.5	70.2	94.7	88.2	84.5
24-task CA-MTL$_{\text{BERT-LARGE}}$	1.12×	5.6%	63.8	86.3/86.0	92.9	93.4	88.1	84.5	94.5	90.3	86.6
RoBERTa LARGE Models — Dev set Results											
RoBERTa-LARGE[4] (Median 5 runs)***	9×	100%	68.0	90.2	90.9	94.7	**	86.6	96.4	**	–
ST RoBERTa-LARGE	9×	100%	68.3	89.2/88.9	92.6	94.8	84.6	87.0	96.4	91.7	88.2
24-task CA-MTL$_{\text{RoBERTa-LARGE}}$	1.12×	5.6%	69.7	89.4/89.3	93.9	94.9	88.8	91.0	96.2	91.0	89.4

B.1.5 Some Results on layer Freezing and with Full Block Attention.

All experiments in this section were run for only 5 epochs, exclusively on the GLUE dataset for the large BERT-based 8-task CA-MTL model. Results in Table B.4 reveal that as we freeze more layers, performance tends to decrease. However, since we wanted to preserve as much pretrained knowledge as possible, we chose to keep at least 50% of layers frozen. While this has slightly lowered our performance on 9 GLUE tasks, we believe that keeping as much of the original pretrained weights is beneficial when increasing the total number of tasks in MTL to 24 or more tasks. However, we did not explore this hypothesis more.

Table B.4 **8-task CA-MTL$_{\text{BERT-LARGE}}$** (see section 3.4.3) for various layer freezing configurations. F1 scores are reported for QQP/MRPC, Spearman's correlation for STS-B, accuracy on the matched/mismatch sets for MNLI, Matthew's correlation for CoLA and accuracy for other tasks. FBA = Full Block Attention

Method	% frozen	# tasks	GLUE								
	layers	g.e ST	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg
LARGE Models — Dev set Results											
ST BERT-LARGE (ours)	0%	—	63.6	86.5/86.0	91.4	91.0	88.5	70.2	93.1	88.2	84.3
CA-MTL	0%	7	60.2	86.2/86.0	92.0	91.5	88.7	76.3	93.3	89.5	84.9
CA-MTL	25%	6	63.7	86.1/85.8	89.1	91.2	88.6	79.7	92.9	88.5	85.1
CA-MTL	50%	3	63.2	85.5/85.5	91.8	90.9	88.3	81.4	93.0	90.1	85.5
CA-MTL FBA	50%	0	60.2	81.7/81.1	88.0	85.8	85.7	78.7	88.6	87.1	81.8

B.1.6 Dataset Description

The datasets that were used for the domain adaptation experiments were SciTail[1] and SNLI[2]. We *jointly* trained a CA-MTL$_{\text{RoBERTa-LARGE}}$ model on 9 GLUE tasks, 8 Super-GLUE[3] tasks, 6 MRQA[4] tasks, and on WNUT2017[5] [156].

Table B.5 GLUE [16] dataset description.References: [1][17], [2][18], [3][19], [4][20], [5][21], [6][16], [7][22]

Acronym	Corpus	|Train|	Task	Domain
CoLA[1]	Corpus of Linguistic Acceptability	8.5K	acceptability	miscellaneous
SST-2[2]	Stanford Sentiment Treebank	67K	sentiment detection	movie reviews
MRPC[3]	Microsoft Research Paraphrase Corpus	3.7K	paraphrase detection	news
STS-B[4]	Semantic Textual Similarity Benchmark	7K	textual similarity	miscellaneous
QQP	Quora Question Pairs	364K	paraphrase detection	online QA
MNLI[5]	Multi-Genre NLI	393K	inference	miscellaneous
RTE[6]	Recognition Textual Entailment	2.5K	inference/entailment	news, Wikipedia
WNLI[7]	Winograd NLI	634	coreference	fiction books

All GLUE tasks are binary classification, except STS-B (regression) and MNLI (three classes). We used the same GLUE data preprocessing as in [6].

Table B.6 Super-GLUE [23] dataset description. References: [1][24], [2][25], [3][26], [4][27], [5][28], [6][23], [7][29], [8][22]

Acronym	Corpus	|Train|	Task	Domain
BoolQ[1]	Boolean Questions	9.4K	acceptability	Google queries, Wikipedia
CB[2]	CommitmentBank	250	sentiment detection	miscellaneous
COPA[3]	Choice of Plausible Alternatives	400	paraphrase detection	blogs, encyclopedia
MultiRC[4]	Multi-Sentence Reading Comprehension	5.1K	textual similarity	miscellaneous
ReCoRD[5]	Reading Comprehension and Commonsense Reasoning	101K	paraphrase detection	news
RTE[6]	Recognition Textual Entailment	2.5K	inference	news, Wikipedia
WiC[7]	Word-in-Context	6K	word sense disambiguation	WordNet, VerbNet
WSC[8]	Winograd Schema Challenge	554	coreference resolution	fiction books

Table B.7 MRQA [30] dataset description. References: [1][31], [2][32], [3][33], [4][34], [5][35], [6][36]

Acronym	Corpus	|Train|	Task	Domain
SQuAD[1]	Stanford QA Dataset	86.6K	crowdsourced questions	Wikipedia
NewsQA[2]	NewsQA	74.2K	crowdsourced questions	news
TriviaQA[3]	TriviaQA	61.7K	trivia QA	web snippets
SearchQA[4]	SearchQA	117.4K	Jeopardy QA	web snippets
HotpotQA[5]	HotpotQA	72.9K	crowdsourced questions	Wikipedia
Natural Questions[6]	Natural Questions	104.7K	search logs	Wikipedia

[1] https://allenai.org/data/scitail; Leaderboard can be found at: https://leaderboard.allenai.org/scitail/submissions/public

[2] https://nlp.stanford.edu/projects/snli/

[3] https://super.gluebenchmark.com/tasks

[4] https://github.com/mrqa/MRQA-Shared-Task-2019

[5] https://github.com/leondz/emerging_entities_17

SuperGLUE has a more diverse task format than GLUE, which is mostly limited to sentence and sentence-pair classification. We follow the same preprocessing procedure as in [23]. All tasks are binary classification tasks, except CB (three classes). Also, WiC and WSC are span based classification tasks. We used the same modified MRQA dataset and preprocessing steps that were used in [257]. All MRQA tasks are span prediction tasks which seeks to identify start and end tokens of an answer span in the input text.

Table B.8 SNLI [37] and SciTail [38] datasets description.

Acronym	**Corpus**	**|Train|**	**Task**	**Domain**
SNLI[1]	Stanford Natural Language Inference	550.2k	inference	human-written English sentence pairs
SciTail[2]	Science and Entailment	23.5K	entailment	Science question answering

SNLI is a natural inference task where we predict three classes. Examples of three target labels are: Entailment, Contradiction, and Neutral (irrelevant). SciTail is a textual entailment dataset. The hypotheses in SciTail are created from multiple-choice science exams and the answer candidates (premise) are extracted from the web using information retrieval tools. SciTail is a binary true/false classification tasks that seeks to predict whether the premise entails the hypothesis. The two datasets are used only for domain adaptation in this study (see section B.1.2 for the details of our approach).

APPENDIX C INTERACTIVE CHAIN PROMPTING

C.1 Appendix: Interactive Chain Prompting

The appendix contains more information on *InterCPt*. We examine limitations of our work in Section C.1.1. In Section C.1.2, we further link the specific prompts to each interactive step in Figure 4.1. In Section C.1.3, we discuss the link between *InterCPt* and methods such as *Chain-of-Thought* and *Least-to-Most prompting.* We discuss other meaningful related work in Section **??**. In Section C.1.4, we provide details on the datasets that we have created such as (1) data statistics and (2) tools, process and pseudocode to create the data. Finally, in Section C.1.7, we list all of the pseudocodes for prompting PaLM for both the User LM and the Translator LM.

C.1.1 Limitations

Our work is about solving query ambiguities in translation which is a relatively unexplored area. Solving unambiguous sentences in Translation is a topic that is most traditionally researched in Translation. During initial experimentation, PaLM was able to correctly detect ambiguous and unambiguous queries in 98% of examples (with a 1,000 sample size and a balanced split between ambiguous/unambiguous labels). Nonetheless, we have not fully explored performance on unambiguous queries and this could be a possible limitation.

It must be noted however that our method is orthogonal to contemporaneous context-less or interaction-less translation work such as Prompting PaLM for Translation (*POMP*) [173] in which prompts, exemplars and instructions are optimized to reach state-of-the-art translation BLEU/BLEURT scores on common WMT benchmarks with unambiguous text (see Related Works Section 4.3 for more details). *InterCPt* without context is equivalent to the *LLMnoExtra* baseline since it uses the same prompt exemplars and the same model without context and without answers from the simulated user (see Section 4.4).

Our paper tackles the issue of user query ambiguities where we assume that the user has background information. For example, if a user wants to translate “are you sure it is pretty?”, the user should know what “it” is and who “you” is. If the user refuses to answer questions, we can default translations to *LLMnoExtra* which is the same as *InterCPt* without context or interaction.

While we have covered more ambiguities across more languages than other prior work, there is still ambiguities and languages that we have not yet tested. This could be another limitation for ambiguities that are significantly different than the ambiguities discussed in our paper. It must be noted that we have chosen common sentence-level ambiguities and that we have left paragraph-level ambiguities for future work. For example, "lexical cohesion" is an ambiguity type that is more common at the paragraph level and *InterCPt* may not detect such ambiguities.

C.1.2 More details on InterCPt interactive steps and links to prompts

To make link between interaction steps in Figure 4.1, the process overview in Section 4.2, the appendix code and templates, we add the following:

Step 1: The Translation LM asks a question on ambiguity using language specific methods in Apppendix C.1.9. It takes as input the English text to Translate *en_text* and outputs the question Q. For example, if we want to translate English to Spanish with a generalist template, we can use *spanish_generalist_translator_interactive(...)*.

Step 2: The User LM answers the question Q generated in step 1 using any method in Appendix C.1.8. It takes as input *en_text* and the context C (ctx in the code) and outputs the answer U.

Step 3: If no other ambiguity is detected, the Translation LM translates using language specific methods in Appendix C.1.9. It takes as input the English text to Translate *en_text*, the question Q, and the answer U and outputs the translation A.

C.1.3 Link with Chain-of-Thought and Least-to-Most prompting

In this section, we add a few more words on the link between *InterCPt* and Chain-of-Thought (CoT) or Least-to-Most (L2M) prompting. CoT performs better than the baseline that has access to the whole information in the problem statement (similar to having context). The behavior is attributed to the sequential solving of subproblems (in our case ambiguity) and a multistep computation (in our case interaction). *LLMwCxt* has access to more information but does not involve multiple computation steps to solve a subproblem. This is how *InterCPt* is most similar to CoT since *InterCPt* uses multistep computation.

C.1.4 More details on AmbigMT ambiguity datasets

In this section, we provide additional information on what the datasets contain and how they were created. As mentioned in Section 4.1, we did not find datasets that covered multiple ambiguities for multiple language pairs. We provide an overview of publicly available datasets in Table C.1. Upon manual inspection of samples from other public datasets, we found that translation queries were often ($> 50\%$) unambiguous since the translation query contained enough information and did not need to rely on the provided context. We inspected 200 samples from *AmbigMT* and found that only ∼3% of queries did not need context to disambiguate the linguistic phenomena.

Source	**Lang Pairs**	**Linguistic Phenomena**	**Test Data**
[258]	en→de	(1) "it" pronoun resolution	12,000
[259]	en→fr	(1) Anaphora resolution, (2) lexical cohesion	900
[260]	en→ru	(1) Ellipsis, (2) lexical cohesion	6,000
[260]	de→en zh→en en→ru	(1) "it" pronoun resolution, (2) lexical cohesion	6,090
AmbigMT	en→es en→fr en→de en→ja	(1) "it" pronoun resolution, (2) gender neutral names (3) neutral professions, (4) polysemy, (5) formality	17,200

Table C.1 Other MT datasets that contain specific linguistic phenomena and provide context. en = English, de = German, fr = French, ru = Russian, zh = Mandarin Chinese, ja = Japanese.

C.1.5 Dataset statistics

We present in Table C.2 the data statistics for *AmbigMT*. For polysemy, the total senses per word is the number of different definitions or meanings found for a specific source English word. Each ambiguity is well balanced across classes formal/informal or feminine/masculine. The Neutral Professions dataset is derived from the Translated Wikipedia Biographies dataset[1] that only covers {en-es, en-de} language pairs.

Table C.2 *AmbigMT* data statistics of each type of class and language pair.
Form = formal, Inform = informal, Mas = Masculine, Fem = Feminine, res = resolution, Prof = Profession.

Lang Pair	**Total Samples**	**Polysemy**	**Formality**		**"it" res.**		**Neutral Names**		**Neutral Prof.**	
		Senses	Form.	Inform.	Mas.	Fem.	Mas.	Fem.	Mas.	Fem.
en→es	4600	3.6	49%	51 %	50%	50%	51%	49%	52%	48%
en→de	4600	3.1	50%	50 %	52%	48%	50%	50%	53%	47%
en→fr	4000	3.3	49%	51 %	50%	50%	51%	49%	—	—
en→ja	4000	3.0	50%	50 %	52%	48%	53%	47%	—	—

[1] https://ai.googleblog.com/2021/06/a-dataset-for-studying-gender-bias-in.html

C.1.6 AmbigMT data creation tools, process and heuristics

In this section, we present the steps, tools and heuristics used to detect ambiguities. For polysemy, formality, "it" resolution, gender neutral names, we extract the data from OpenSubtitles corpora and neutral professions from Translated Wikipedia Biographies. The source data that was used consists of parallel sentence level pairs. We first detect a sentence that has a specific ambiguity and extract the context by taking three to five preceding English sentences, depending on sentence size. For Polysemy, the context is an English sentence that contains the polysemous word that will be translated. The code and datasets are released **here**.

Polysemy

We provide the following list of steps to create the polysemy dataset for all languages:

1. Extract polysemous words from Wordnet. [261] using the NLTK toolkit [262][2].
    - Create a list of English words.
    - Compute the number of definitions per word without counting definitions with synonym overlap.
    - Extract polysemous words (w_e) with more than three definitions and a word length greater than four.
2. For each Polysemous English word w_e, extract a list $l_x = \{w_{x1}, \dots, w_{xN}\}$ of possible word translations using the Google Cloud Translation v2 API, where $x \in \{\mathrm{es}, \mathrm{fr}, \mathrm{de}, \mathrm{ja}\}$ is the target language.
3. For each Polysemous English word w_e and each target language $x \in \{\mathrm{es}, \mathrm{fr}, \mathrm{de}, \mathrm{ja}\}$:
    - Find a sentence that contains the word w_e in the OpenSubtitle dataset.
    - If the parallel sentence contains one of the translated word $w_{xi} \in l_x$ from step 2 and no other translated word, keep the English sentence as context.

Formality

Each language has specific formality rules. For Japanese, we direct the reader to our public code: `https://github.com/jpilaul/interactive_chain_prompting`. We provide the following list of steps to create the formality dataset for Spanish, French and German:

1. Find a sentence that contains "you" or "your" and that has word count less than 20, in the English OpenSubtitle corpus.

[2] See example in `https://www.nltk.org/howto/wsd.html`

2. Select parallel sentences for each target language $x \in \{\mathrm{es}, \mathrm{fr}, \mathrm{de}, \mathrm{ja}\}$ that meet the following criteria.
3. If $x ==$ es, check the following in parallel Spanish sentence (all checks are initialized to FALSE):
    - If all verbs finish by “s”, “ste” or “os”, then is_verb_informal = TRUE.
    - If any pronouns is “usted”, then is_pronoun_formal = TRUE.
    - If any pronouns is in [“tú”,“tu”,“te”, “vos”, “vosotros”], then is_pronoun_informal = TRUE.
    - If any determinants is “su”, then is_determinant_formal = TRUE.
    - If any determinants is in [“tu”,“vosotros”, “vosotras”] then is_determinant_informal = TRUE.
    - is_informal = is_verb_informal and is_pronoun_informal and is_determinant_informal.
    - is_formal = is_pronoun_formal and is_determinant_formal.
4. If $x ==$ fr, check the following in parallel French sentence (all checks are initialized to FALSE):
    - If any verbs finish by “x”, “s” or “ons”, then is_verb_informal = TRUE.
    - If any verbs finish by “ez”, then is_verb_formal = TRUE.
    - If one of the pronouns is “vous”, then is_pronoun_formal = TRUE.
    - If one of the pronouns is “tu”, then is_pronoun_informal = TRUE.
    - If one of the determinants is in [“vos”,“votre”], then is_determinant_formal = TRUE.
    - If one of the determinants is in [“tes”,“ton”, “ta”, “toi”] then is_determinant_informal = TRUE.
    - is_informal = is_verb_informal and is_pronoun_informal and is_determinant_informal.
    - is_formal = is_verb_formal and is_pronoun_formal and is_determinant_formal.
5. If $x ==$ de, check the following in parallel German sentence (all checks are initialized to FALSE):
    - If “!” not in sentence and one of the pronouns is in [“Sie”,“Ihr”, “Ihre”, “Ihren”, “Ihrem”, “Ihrer”, “Ihres”], then is_pronoun_formal = TRUE.
    - If one of the pronouns is in [“du”,“dein”, “deine”, “deinen”, “deinem”, “deiner”, “deines”, “dich”], then is_pronoun_formal = TRUE.
    - If “!” in sentence one of the pronouns is in [“er”,“sie”, “es”, “ihr”], then is_pronoun_formal = TRUE.
    - is_informal = is_pronoun_informal.
    - is_formal = is_pronoun_formal.

6. Keep samples if is_formal != is_informal, use ‘formal’ label if is_formal or ‘informal’ label if is_informal.
7. For each sample, create context by keeping the preceding three to five English sentences, depending if word count is above 20.

“it” resolution

We provide the following list of steps to create the “it” resolution dataset. The steps apply to all languages:

1. For each English sentence in the OpenSubtitle dataset, keep sentences where the word“it” exists.
    - Using a dependency parser, if “it” is expletive[3], skip sample.
    - In the parallel Spanish, French, German or Japanese sentence, if the sentence does not contain a verb and a gendered pronouns, skip sample.
    - Keep gender label.
2. For each sample, create context by keeping the preceding three to five English sentences, depending if word count is above 20.

Gender Neutral Names

We provide the following list of steps to create the gender neutral names dataset. Please note that for simplicity we used binary genders. Genders beyond female and male will be left for future work. The steps apply to all languages:

1. Compile a list L_{gnn} of gender neutral (unisex) names
    - Collect a list of names with gender statistic such as the percentage of people with the name who identify as female or male[4].
    - Keep the names that are used in approximately equal proportions (unisex) with at least a female or male proportion above 40%.
2. For each gender neutral name $\in L_{gnn}$, find a sentence that contains the name in the English sentence and keep the corresponding parallel sentence in Spanish, French, German or Japanese.
    - If the English sentence has gendered pronouns, skip the sentence if multiple genders are detected.

[3] The spaCy dependency parser can be used to find expletive “it”.

[4] Names with gender statistics were compiled and combined using a Japanese names database [263] and a English names database that originates from the United States Social Security Administration.

- If the English sentence has no gendered pronouns, use a Part-of-Speech tagger[5] on the corresponding parallel sentence in Spanish, French, German or Japanese and skip the sentence if multiple genders are detected.
- Keep gender label.

3. Replace gendered pronouns with [pr] in the source English sentence to remove simple clues about the name's gender.
4. For each sample, create context by keeping the succeeding three to five English sentences, depending if word count is above 20.

C.1.7 Prompt templates used in experiments

In this section, we discuss the main prompt templates used in experiments. This includes *InterCPt Translator* generalist and specialist templates to ask questions about ambiguities and exemplars to translate in French, Spanish, German or Japanese. It also includes *InterCPt User* generalist and specialist templates to answer questions given a context. We also provide the prompt templates for the PaLM-with-Context experiments where we use context and the same exemplars to translate in French, Spanish, German or Japanese. Please note that we have normalized special characters for simplicity. The German and Japanese templates as well as Spanish and French templates with special characters can be found in our public code and data repository. In the python methods listed below, *en_text* is the input query, *ctx* is the context, *question* is the question from the Translator model and *anwer* is the answer from the User model.

C.1.8 InterCPt Simulated User Prompts

The 8-shot generalist Simulated *User* prompt template is the same for all languages and is provided in code block listing C.1.

```
def generalist_simulated_user_context(en_text, question, ctx):
    """Generalist Simulated user has access to context and answers the question."""

    templated_input =
f"""[web] Given a Context (C), provide an Answer (A) to the Question (Q):

S: about
```

[5] Language specific spaCy models could be used.

```
C: About 2% of the households are enumerated using the canvasser
   method.
Q: Is "about" an adverb that means approximately, near or a
   preposition that means regarding, over, surrounding?
A: "about" means approximately.


S: rent
C: Many single women cannot live independently because they cannot (
   afford to) own or rent housing
Q: Is "rent" a tenant's regular payment for a property or to pay
   someone for the use of something?
A: "rent" is to pay someone for the use of something.


S: abstract
C: For the international community is not an abstract concept, it
   consists of us ourselves.
Q: Is "abstract" to consider theoretically, to extract something, or
    a summary, or an adjective?
A: "abstract" is an adjective that modifies "concept" in the phrase
   "abstract concept".


S: What do you mean?
C: Daria, I just think that your field of vision could really be
   enhanced... - Come on, Mom. - It's not my field of vision you
   want to enhance.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?
A: "you" is 'informal' since the listener is the speaker's "mom", it
    implies a familiarity with the listener "you".


S: This will accelerate your metabolic functions-- help you make the
    transition.
C: At the very least, get them to hold their fire. - Captain, the
   transporters are off-line. The docking port hasn't been hit yet.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?
```

```
A: "you" is 'formal' since "you" refers to a Captain and the speaker will typically use a polite form.


S: You know where it begins, you never know where it ends...
C: Someone once told me we always are where we're supposed to be. - Now I believe it. - Life is a journey.
Q: "you" can be neutral, formal, informal. Who does "you" refer to in (S)?
A: "you" is \'neutral\' because it is a generic "you" that refers to people in general on their journey through life.


S: it is also very pretty.
C: Even when it is pouring outside, this umbrella is both practical and elegant.
Q: What does "it" refer to?
A: "it" is a harp.


S: Tell me, why do they have to tilt it?
C: -Frog is wrong. - I see here that you play the harp.
Q: What does "it" refer to?
A: "it" is an umbrella.


S: {en_text.strip()}
C: {ctx.strip()}
Q: {question}
A:"""
    return templated_input
```

Pseudocode C.1 *InterCPt* Generalist Simulated User Prompt Template

The 8-shot *formality* specialist Simulated *User* prompt template is the same for all languages and is provided in code block listing C.2.

```
def formality_simulated_user_context(en_text, question, ctx):
    """Formality simulated user has access to context and answers the question."""

    templated_input =
f"""[web] Given a Context (C), provide an Answer (A) to the Question (Q) about Sentence (S):

S: This is for you, too.
C: I'm Freya. - Welcome to Denmark, Mr. Helm. - You always greet people like this? - I'm Freya Carlson, your Tourist Bureau contact.
Q: "you" can be neutral, formal, informal. Who does "you" refer to in (S)?
A: "you" is \'formal\' since "you" refers to a customer or tourist that Freya Carlson is greeting with the polite form "Mr.".


S: - i can gladly help you.
C: I will go to town to fetch the materials. Once I return, we can repair your majesty's royal carriage.
Q: "you" can be formal or informal. Who does "you" refer to?
A: "you" is \'formal\' since "you" refers to "your majesty".


S: You know what I mean.
C: Elizabeth, will you bring the binoculars? - [Elizabeth] Mm, the stench is horrible. [John] Here, take a hold of this. - [Elizabeth] Is it dead?
Q: "you" can be neutral, formal, informal. Who does "you" refer to in (S)?
A: "you" is \'informal\' since the listener "John" has familiarity with the speaker and uses the first name "Elizabeth".


S: You think you can make it through that kind of stuff, you think you can make it through anything.
C: Well, transitions are hard. - Been together ever since college. - Been through a lot. - You know, us coming out to her family, and her brother dying.
Q: "you" can be neutral, formal, informal. Who does "you" refer to
```

```
   in (S)?
A: "you" is \'neutral\' because it is a generic "you" that refers to
    people in general going through a difficult moment.


S: You can imagine the princess-sized tantrum that followed.
Q: "you" can be neutral, formal, informal. Who does "you" refer to
   in (S)?
C: This is the bike that I learned to ride on. - I just didn't know
   my mom kept it. - It used to have these training wheels on the
   back with lights that would flash every time you pedaled. - Then
   one day, my mom took them off and said it was time to be a big
   girl.
A: "you" is \'informal\' since the speaker is talking about a funny
   childhood memory which implies a familiarity with the listener "
   you".


S: Can I just say, it's been an absolute pleasure to finally meet
   you?
C: Generations of Daleks just woke up very cross, and they're coming
    up the pipes. - Or to put it another way... bye! - Doctor, you
   must help me.
Q: "you" can be neutral, formal, informal. Who does "you" refer to
   in (S)?
A: "you" is \'formal\' since "you" refers to a "Doctor" that the
   speaker just met.


S: You know where it begins, you never know where it ends...
C: Someone once told me we always are where we're supposed to be. -
   Now I believe it. - Life is a journey.
Q: "you" can be neutral, formal, informal. Who does "you" refer to
   in (S)?
A: "you" is \'neutral\' because it is a generic "you" that refers to
    people in general on their journey through life.


```

```
S: City policemen questioned many of you this week.
C: Lying on his belly, he was carried home on a makeshift stretcher. - Next Sunday, after the service, the Baron asked the pastor to let him speak.
Q: "you" can be neutral, formal, informal. Who does \"you\" refer to in (S)?
A: "you" is \'formal\' since the speaker directly addresses several people or "many of you", the plural form of "you".


S: {en_text.strip()}
C: {ctx.strip()}
Q: {question}
A: """
    return templated_input
```

Pseudocode C.2 *InterCPt* **Formality** Specialist Simulated User Prompt Template

The 8-shot *polysemy* specialist Simulated *User* prompt template is the same for all languages and is provided in code block listing C.3.

```
def polysemy_simulated_user_context(en_text, question, ctx):
    """Polysemy simulated user has access to context and answers the question."""

    templated_input =
f"""[web] Given a Context (C), provide an Answer (A) to the Question (Q):

S: abstract
C: For the international community is not an abstract concept, it consists of us ourselves.
Q: Is "abstract" to consider theoretically, to extract something, or a summary, or an adjective?
A: "abstract" is an adjective that modifies the word "concept".


S: abstract
C: We need to abstract the data from various studies.
Q: Is "abstract" to consider theoretically, to extract something, or
```

```
a summary, or an adjective?
A: "abstract" means to extract something.


S: about
C: About 2% of the households are enumerated using the canvasser method.
Q: Is "about" an adverb that means approximately, near or a preposition that means regarding, over, surrounding?
A: "about" means approximately.


S: about
C: The story is about soldier returning home after the war.
Q: Is "about" an adverb that means approximately, near or a preposition that means regarding, over, surrounding?
A: "about" means regarding.


S: bank
C: The online banking application does not work. I tried a few times and I could not transfer the funds. I went to the bank.
Q: Is "bank" a financial institution, the edge of a river, a set or series of similar things or the cushion of a pool?
A: "bank" is a financial institution.


S: rent
C: Many single women cannot live independently because they cannot (afford to) own or rent housing
Q: Is "rent" a tenant's regular payment for a property or to pay someone for the use of something?
A: "rent" is to pay someone for the use of something.


S: bat
C: The bat flew over the forest and back to its cave.
Q: Is "bat" an animal or a sports equipment?
```

```
A: "bat" is an animal.


C: {ctx}
Q: {question}
A: """
    return templated_input
```

Pseudocode C.3 *InterCPt* **Polysemy** Specialist Simulated User Prompt Template

C.1.9 InterCPt Generalist Prompt Templates for each target language

The 8-shot *Spanish* generalist *Translator* prompt template is the same for all test ambiguity data and is provided in code block listing C.4.

```
def spanish_generalist_translator_interactive(en_text, question=None, answer=None):
    """Translation model asks questions and uses answers to translate"""
    if answer == None:
        # Ask questions
        instructions = "[web] Given sentence 'S' to translate to Spanish, ask clarifying questions 'Q' to clarify ambiguities or multiple senses:"
    else:
        # Translate given answer
        instructions = "[web] Given answer 'U' to question 'Q', provide the Spanish translation 'A' of sentence 'S'. Provide the best answer:"

    templated_input =
"""

S: about
Q: Is "about" an adverb that means approximately, near or a preposition that means regarding, over, surrounding?%s


S: rent
```

```
Q: Is "rent" a tenant's regular payment for a property or to pay someone for the use of something?%s


S: abstract
Q: Is "abstract" to consider theoretically, to extract something, or a summary, or an adjective?%s


S: You think if I get contacts I'll suddenly turn into the homecoming queen.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: This will accelerate your metabolic functions-- help you make the transition.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: They could wait 'till you're on the beach, then cut loose, or start firing right away.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: can't they just build it on an angle?
Q: What does "it" refer to?%s


S: It is also very pretty.
Q: What does "it" refer to?%s


"""
    if answer is None:
        templated_input = templated_input % ('', '', '', '', '', '',
    '', '')
        templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ:"
    else:
        templated_input = templated_input % (
            '\nU: "about" means approximately.\nA: aproximadamente, cerca de, alrededor de, casi, mas o menos',
            '\nU: "rent" is to pay someone for the use of something.\nA: alquilar, arrendar, rentar',
            '\nU: "abstract" is an adjective that modifies "concept" in the phrase "abstract concept".\nA: abstraccion, abstracto',
            '\nU: "you" is \'informal\' since the listener is the speaker\'s "mom", it implies a familiarity with the listener "you".\nA: Tu piensas que si uso lentes de contacto de repente me convertire en la nueva reina del colegio.',
            '\nU: "you" is \'formal\' since "you" refers to a Captain and the speaker will typically use a polite form.\nA: Esto acelerara sus funciones metabolicas. Lo ayudara a hacer la transicion.',
            '\nU: "you" is \'neutral\' because it is a generic "you" that refers to people in general and not someone specific.\nA: Podian aguardar a que uno estuviera en la playa y atacar o comenzar a disparar.',
            '\nU: "it" is a harp.\nA: no pueden hacerla en angulo?',
            '\nU: "it" is an umbrella.\nA: Es muy bonita tambien.',
        )
    templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ: {question}\nU: {answer}\nA: "
    return templated_input
```

Pseudocode C.4 *InterCPT* **Spanish** Generalist Translator Prompt Template

The 8-shot *French* generalist *Translator* prompt template is the same for all test ambiguity data and is provided in code block listing C.5.

```
def french_generalist_translator_interactive(en_text, question=None, answer=None):
    """Translation model asks questions and uses answers to translate"""
    if answer == None:
        # Ask questions
        instructions = "[web] Given sentence ’S’ to translate to French, ask clarifying questions ’Q’ to clarify ambiguities or multiple senses:"
    else:
        # Translate given answer
        instructions = "[web] Given answer ’U’ to question ’Q’, provide the French translation ’A’ of sentence ’S’. Provide the best answer:"

    templated_input = """

S: about
Q: Is "about" an adverb that means approximately, near or a preposition that means regarding, over, surrounding?%s


S: rent
Q: Is "rent" a tenant’s regular payment for a property or to pay someone for the use of something?%s


S: abstract
Q: Is "abstract" to consider theoretically, to extract something, or a summary, or an adjective?%s


S: You know where it begins, you never know where it ends...
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: This is for you, too.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: You know where it begins, you never know where it ends...
```

```
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: I'll help you find it before [pr] does.
Q: What does "it" refer to?%s


S: [pr] must have forced it somehow.
Q: What does "it" refer to?%s


"""

    if answer is None:
        templated_input = templated_input % ('', '', '', '', '', '', '', '')
        templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ:"
    else:
        templated_input = templated_input % (
        '\nU: "about" means approximately.\nA: environ, presque, quelque, a peu pres, approximativement',
        '\nU: "rent" is to pay someone for the use of something.\nA: louer',
        '\nU: "abstract" is an adjective that modifies "concept" in the phrase "abstract concept".\nA: abstraction, abstrait',
        '\nU: "you" is \'informal\' since the speaker has familiarity with the listener and uses the first name "Jerry".\nA: A qui as-tu parle ?',
        '\nU: "you" is \'formal\' since "you" refers to a customer or tourist that Freya Carlson is greeting with the polite form "Mr.".\nA: Ceci est pour vous.',
        '\nU: "you" is \'neutral\' because it is a generic "you" that refers to people in general going through a difficult moment.\nA: On sait ou cela commence, mais on ne sait jamais ou cela se termine...',
        '\nU: "it" is a key.\nA: Je vous aiderai a la trouver avant
elle.’,
        ’\nU: "it" is a gate.\nA: Il a du le forcer d\’une maniere ou d\’une autre.’,
        )
    templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ: {question}\nU: {answer}\nA: "
    return templated_input
```

Pseudocode C.5 *InterCPt* **French** Generalist Translator Prompt Template

C.1.10 InterCPt Specialist Prompt Templates for each target language

The *Spanish formality* specialist *Translator* prompt template is the same for all test ambiguity data and is provided in code block listing C.6.

```
def spanish_formality_translator_interactive(en_text, question=None, answer=None):
    """Translation model asks questions and uses answers to translate"""
    if answer == None:
        #  Ask questions
        instructions = "[web] Given sentence ’S’ to translate to Spanish, ask clarifying questions ’Q’ to clarify ambiguities or multiple senses:"
    else:
        #  Translate given answer
        instructions = "[web] Given answer ’U’ to question ’Q’, provide the Spanish translation ’A’ of sentence ’S’. Provide the best answer:"

    templated_input = """

S: This will accelerate your metabolic functions-- help you make the transition.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: Poor baby... here’s yours!
```

```
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: They could wait ’till you’re on the beach, then cut loose, or start firing right away.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: You think if I get contacts I’ll suddenly turn into the homecoming queen.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: For centuries, we have watched you, listened to your radio signals and learned your speech and your culture.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: I never have. I’m not sure you’re supposed to.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


"""

    if answer is None:
        templated_input = templated_input % (’’, ’’, ’’, ’’, ’’, ’’)
        templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ:"
    else:
        templated_input = templated_input % (
        ’\nU: "you" is \’formal\’ since "you" refers to a Captain and the speaker will typically use a polite form.\nA: Esto acelerara sus funciones metabolicas. Lo ayudara a hacer la
transicion.’,
        ’\nU: "you" is \’informal\’ since the speaker has familiarity with the listener and they both use "baby" and "buddy" to address each other.\nA: Pobre bebe... aqui esta el tuyo!’,
        ’\nU: "you" is \’neutral\’ because it is a generic "you" that refers to people in general and not someone specific.\nA: Podian aguardar a que uno estuviera en la playa y atacar o comenzar a disparar.’,
        ’\nU: "you" is \’informal\’ since the listener is the speaker\’s "mom", it implies a familiarity with the listener "you".\nA: Tu piensas que si uso lentes de contacto de repente me convertire en la nueva reina del colegio.’,
        ’\nU: "you" is \’formal\’ since the speaker addresses people not acquainted with or unfamiliar.\nA: Durante siglos, los hemos observado, escuchado sus senales de radio. Hemos aprendido su idioma y cultura.’,
        ’\nU: "you" is \’neutral\’ because it is a generic "you" that refers to people in general that have been in this "line of work".\nA: Yo no. No creo que uno deba acostumbrarse.’
        )
    templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ: {question}\nU: {answer}\nA: "
    return templated_input
```

Pseudocode C.6 *InterCPt* **Spanish Formality** Specialist Translator Prompt Template

The *Spanish polysemy* specialist *Translator* prompt template is the same for all test ambiguity data and is provided in code block listing C.7. Please note that the instructions for the translation step is different than the generalist or the formality specialist template.

```
def spanish_polysemy_translator_interactive(en_text, question=None, answer=None):
    """Translation model asks questions and uses answers to translate"""
    if answer == None:
        #  Ask questions
        instructions = "[web] Given an English word ’S’ to translate to Spanish, to clarify ambiguities and understand multiple senses ask questions ’Q’:"
    else:
        #  Translate given answer
        instructions = "[web] Given answer ’U’ to question ’Q’, Translate word ’S’ into Spanish and provide unique and non-repeating synonyms in ’A’:"

    templated_input = """

S: abstract
Q: Is "abstract" to consider theoretically, to extract something, or a summary, or an adjective?%s


S: abstract
Q: Is "abstract" to consider theoretically, to extract something, or a summary, or an adjective?%s


S: about
Q: Is "about" an adverb that means approximately, near or a preposition that means regarding, over, surrounding?%s


S: bank
Q: Is "bank" to tilt sideways, or a financial institution, the edge of a river, a set or series of similar things or the cushion of a pool?%s


S: rent
Q: Is "rent" a tenant’s regular payment for a property or to pay someone for the use of something?%s


"""

    if answer is None:
        templated_input = templated_input % (’’, ’’, ’’, ’’, ’’)
        templated_input = f"{instructions}\n" + templated_input + f"
```

```
S: {en_text}\nQ: "
    else:
        templated_input = templated_input % (
        '\nU: "abstract" is an adjective that modifies "concept" in
the phrase "abstract concept".\nA: abstraccion, abstracto',
        '\nU: "abstract" means to extract something.\nA: abstraer',
        '\nU: "about" means approximately.\nA: aproximadamente,
cerca de, alrededor de, casi, mas o menos',
        '\nU: "bank" is a financial institution.\nA: banco',
        '\nU: "rent" is to pay someone for the use of something.\nA:
 alquilar, arrendar, rentar'
        )
    templated_input = f"{instructions}\n" + templated_input + f"S: {
en_text}\nQ: {question}\nU: {answer}\nA: "
    return templated_input
```

Pseudocode C.7 *InterCPt* **Spanish Polysemy** Specialist Translator Prompt Template

The *French formality* specialist *Translator* prompt template is the same for all test ambiguity data and is provided in code block listing C.8.

```
def french_formality_translator_interactive(en_text, question=None,
answer=None):
    """Translation model asks questions and uses answers to
translate"""
    if answer == None:
        #  Ask questions
        instructions = "[web] Given sentence 'S' to translate to
French, ask clarifying questions 'Q' to clarify ambiguities or
multiple senses:"
    else:
        #  Translate given answer
        instructions = "[web] Given answer 'U' to question 'Q',
provide the French translation 'A' of sentence 'S'. Provide the
best answer:"

    templated_input = """

S: This is for you, too.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%
```

```
s


S: To whom have you been talking?
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: You know where it begins, you never know where it ends...
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: You can imagine the princess-sized tantrum that followed.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: City policemen questioned many of you this week.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


S: You think you can make it through that kind of stuff, you think you can make it through anything.
Q: "you" can be neutral, formal, informal. Who does "you" refer to?%s


"""

    if answer is None:
        templated_input = templated_input % (’’, ’’, ’’, ’’, ’’, ’’)
        templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ:"
    else:
        templated_input = templated_input % (
        ’\nU: \nA: Ceci est pour vous.’,
        ’\nU: \nA: A qui as-tu parle ?’,
        ’\nU: \nA: On sait ou cela commence, mais on ne sait jamais ou cela se termine...’,
        ’\nU: \nA: Tu peux imaginer la colere de princesse qui a suivi.’,
        ’\nU: \nA: Les gendarmes sont venus interroger nombre d\’entre vous.’,
        ’\nU: \nA: On pense que quand on arrive a traverser ce genre de chose, on peut traverser n\’importe quoi.’
        )
    templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ: {question}\nU: {answer}\nA: "
    return templated_input
```

Pseudocode C.8 *InterCPt* **French Formality** Specialist Translator Prompt Template

The *French polysemy* specialist *Translator* prompt template is the same for all test ambiguity data and is provided in code block listing C.9. Please note that the instructions for the translation step is different than the generalist or the formality specialist template.

```
def french_polysemy_translator_interactive(en_text, question=None, answer=None):
    """Translation model asks questions and uses answers to translate"""
    if answer == None:
        # Ask questions
        instructions = "[web] Given an English word ’S’ to translate to French, to clarify ambiguities and understand multiple senses ask questions ’Q’:"
    else:
        # Translate given answer
        instructions = "[web] Given answer ’U’ to question ’Q’, Translate word ’S’ into French and provide unique and non-repeating synonyms in ’A’:"

    templated_input = """

S: abstract
Q: Is "abstract" to consider theoretically, to extract something, or a summary, or an adjective?%s
```

```


S: abstract
Q: Is "abstract" to consider theoretically, to extract something, or a summary, or an adjective?%s


S: about
Q: Is "about" an adverb that means approximately, near or a preposition that means regarding, over, surrounding?%s


S: bank
Q: Is "bank" to tilt sideways, or a financial institution, the edge of a river, a set or series of similar things or the cushion of a pool?%s


S: rent
Q: Is "rent" a tenant’s regular payment for a property or to pay someone for the use of something?%s


"""

    if answer is None:
        templated_input = templated_input % (’’, ’’, ’’, ’’, ’’)
        templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ: "
    else:
        templated_input = templated_input % (
        ’\nU: "abstract" is an adjective that modifies "concept" in the phrase "abstract concept".\nA: abstraction, abstrait’,
        ’\nU: "abstract" means to extract something.\nA: abstraire, extraire’,
        ’\nU: "about" means approximately.\nA: environ, presque, quelque, a peu pres, approximativement’,
        ’\nU: "bank" is a financial institution.\nA: banque’,
        '\nU: "rent" is to pay someone for the use of something.\nA: louer'
        )
    templated_input = f"{instructions}\n" + templated_input + f"S: {en_text}\nQ: {question}\nU: {answer}\nA: "
    return templated_input
```

Pseudocode C.9 *InterCPt* **French Polysemy** Specialist Translator Prompt Template

C.1.11 PaLM-with-Context Generalist Prompt Templates for each target language

The 8-shot PaLM-with Context *Spanish* generalist prompt template is the same for all test ambiguity data and is provided in code block listing C.10.

```
def spanish_baseline_generalist_translator_context(en_text, ctx):
    """Translation model uses context to translate."""

    templated_input = f"""[web] Given context 'C', Translate 'T' from English to Spanish:

C: About 2% of the households are enumerated using the canvasser method.
T: about
A: aproximadamente, cerca de, alrededor de, casi, mas o menos


C: Many single women cannot live independently because they cannot (afford to) own or rent housing
T: rent
A: alquilar, arrendar, rentar


C: For the international community is not an abstract concept, it consists of us ourselves.
T: abstract
A: abstraccion, abstracto


```

```
C: Daria, I just think that your field of vision could really be enhanced... - Come on, Mom. - It's not my field of vision you want to enhance. - What do you mean?
T: You think if I get contacts I'll suddenly turn into the homecoming queen.
A: Tu piensas que si uso lentes de contacto de repente me convertire en la nueva reina del colegio.


C: At the very least, get them to hold their fire. - Captain, the transporters are off-line. - The docking port hasn't been hit yet.
T: This will accelerate your metabolic functions-- help you make the transition.
A: Esto acelerara sus funciones metabolicas. Lo ayudara a hacer la transicion


C: Some of the guys got a little sick. - They were scared; I was scared. - I don't think we had any reason to be otherwise.
T: They could wait 'till you're on the beach, then cut loose, or start firing right away.
A: Podian aguardar a que uno estuviera en la playa y atacar o comenzar a disparar.


C: Even when it is pouring outside, this umbrella is both practical and elegant.
T: It is also very pretty.
A: Es muy bonita tambien.


C: -Frog is wrong. - I see here that you play the harp. - Tell me, why do they have to tilt it?
T: can't they just build it on an angle?
A: no pueden hacerla en angulo?


```

```
C: {ctx}
T: {en_text}
A:"""
    return templated_input
```

Pseudocode C.10 PaLM-with-Context **Spanish** Generalist Prompt Template

The 8-shot PaLM-with Context *French* generalist prompt template is the same for all test ambiguity data and is provided in code block listing C.11.

```
def french_baseline_generalist_translator_context(en_text, ctx):
    """Translation model uses context to translate."""

    templated_input = f"""[web] Given context 'C', Translate 'T' from English to French:

C: About 2% of the households are enumerated using the canvasser method.
T: about
A: environ, presque, quelque, a peu pres, approximativement


C: Many single women cannot live independently because they cannot (afford to) own or rent housing
T: rent
A: louer


C: For the international community is not an abstract concept, it consists of us ourselves.
T: abstract
A: abstraction, abstrait


C: I believe! --Who else knows? --I don't know. - Jerry, names! - I don't want to dance!
T: To whom have you been talking?
A: A qui as-tu parle ?


```

```
C: I’m Freya. - Welcome to Denmark, Mr. Helm. - You always greet people like this? - I’m Freya Carlson, your Tourist Bureau contact. - These are for you. Street maps, places of interest.
T: This is for you, too.
A: Ceci est pour vous.


C: It’s like the city’s changed her. - Well, transitions are hard. - Been together ever since college. - Been through a lot. - You know, us coming out to her family, and her brother dying.
T: You know where it begins, you never know where it ends...
A: On sait ou cela commence, mais on ne sait jamais ou cela se termine...


C: Even when it is pouring outside, this umbrella is both practical and elegant.
T: it is also very pretty.
A: il est aussi tres beau.


C: Okay, you don’t smash the cherry on that. Just plop it in at the end.
T: Try to keep it in the top of the glass.
A: Essaie de la garder dans le haut du verre.


C: {ctx}
T: {en_text}
A:"""
    return templated_input
```

Pseudocode C.11 PaLM-with-Context **French** Generalist Prompt Template

C.1.12 PaLM-with-Context Specialist Prompt Templates for each target language

The PaLM-with Context *Spanish Formality* specialist prompt template is the same for all test ambiguity data and is provided in code block listing C.12.

```
def spanish_baseline_formality_translator_context(en_text, ctx):
    """Translation model uses context to translate."""

    templated_input = f"""[web] Given context ’C’, Translate ’T’ from English to Spanish:

C: At the very least, get them to hold their fire. - Captain, the transporters are off-line. - The docking port hasn’t been hit yet.
T: This will accelerate your metabolic functions-- help you make the transition.
A: Esto acelerara sus funciones metabolicas. Lo ayudara a hacer la transicion.

C: Who? - Me! - I think I’ve got a cold. - "Hey buddy, give me a Magic Hug will you!" - Magic Hug! - And me? - Shut up Swami
T: Poor baby... here’s yours!
A: Pobre bebe... aqui esta el tuyo!

C: Some of the guys got a little sick. - They were scared; I was scared. - I don’t think we had any reason to be otherwise.
T: They could wait ’till you’re on the beach, then cut loose, or start firing right away.
A: Podian aguardar a que uno estuviera en la playa y atacar o comenzar a disparar.

C: Daria, I just think that your field of vision could really be enhanced... - Come on, Mom. - It’s not my field of vision you want to enhance. - What do you mean?
T: You think if I get contacts I’ll suddenly turn into the homecoming queen.
A: Tu piensas que si uso lentes de contacto de repente me convertire en la nueva reina del colegio.

C: Men of earth, we of the planet Mars give you this warning. - We have known your planet earth since the first creature crawled out of the primeval slime of your seas to become man.
T: For centuries, we have watched you, listened to your radio
```

```
signals and learned your speech and your culture.
A: Durante siglos, los hemos observado, escuchado sus senales de radio. Hemos aprendido su idioma y cultura.

C: Pull over here. This is the spot. - I guess you run into a lot of dead bodies in your line of work. - You get used to it.
T: I never have. I’m not sure you’re supposed to.
A: Yo no. No creo que uno deba acostumbrarse.

C: {ctx}
T: {en_text}
A:"""
    return templated_input
```

Pseudocode C.12 PaLM-with-Context **Spanish Formality** Specialist Prompt Template

The PaLM-with Context *Spanish Polysemy* specialist prompt template is the same for all test ambiguity data and is provided in code block listing C.13.

```
def spanish_baseline_polysemy_translator_context(en_text, ctx):
    """Translation model uses context to translate."""

    templated_input = f"""[web] Given context ’C’, Translate ’T’ from English to Spanish:


C: Many single women cannot live independently because they cannot (afford to) own or rent housing
T: rent
A: alquilar, arrendar, rentar


C: We need to abstract the data from various studies.
T: abstract
A: abstraer


C: About 2% of the households are enumerated using the canvasser method.
T: about
```

```
A: aproximadamente, cerca de, alrededor de, casi, mas o menos


C: The bat flew over the forest and back to its cave.
T: bat
A: murcielago


C: For the international community is not an abstract concept, it consists of us ourselves.
T: abstract
A: abstraccion, abstracto


C: {ctx}
T: {en_text}
A:"""
    return templated_input
```

Pseudocode C.13 PaLM-with-Context **Spanish Polysemy** Specialist Prompt Template

The PaLM-with Context *French Formality* specialist prompt template is the same for all test ambiguity data and is provided in code block listing C.14.

```
def french_baseline_formality_translator_context(en_text, ctx):
    """Translation model uses context to translate."""

    templated_input = f"""[web] Given context ’C’, Translate ’T’ from English to French:

C: I’m Freya. - Welcome to Denmark, Mr. Helm. - You always greet people like this? - I’m Freya Carlson, your Tourist Bureau contact. - These are for you. Street maps, places of interest.
T: This is for you, too.
A: Ceci est pour vous.

C: I believe! - -Who else knows? - -I don’t know. - Jerry, names! - I don’t want to dance!
T: To whom have you been talking?
A: A qui as-tu parle ?
```

```

C: It's like the city's changed her. - Well, transitions are hard. - Been together ever since college. - Been through a lot. - You know, us coming out to her family, and her brother dying.
T: You know where it begins, you never know where it ends...
A: On sait ou cela commence, mais on ne sait jamais ou cela se termine...

C: You know, if you're gonna go for a spin, I suggest you get your helmet. - This is the bike that I learned to ride on. - I just didn't know my mom kept it. - It used to have these training wheels on the back with lights that would flash every time you pedaled. - Then one day, my mom took them off and said it was time to be a big girl.
T: You can imagine the princess-sized tantrum that followed.
A: Tu peux imaginer la colere de princesse qui a suivi.

C: He was in a state of shock, unable to walk. - Lying on his belly, he was carried home on a makeshift stretcher. - Next Sunday, after the service, the Baron asked the pastor to let him speak.
T: City policemen questioned many of you this week.
A: Les gendarmes sont venus interroger nombre d\'entre vous.

C: I tried to explain... He might have gotten hurt! - I was actually doing him a favour. - Someone once told me we always are where we're supposed to be. - Now I believe it. - Life is a journey.
T: You think you can make it through that kind of stuff, you think you can make it through anything.
A: On pense que quand on arrive a traverser ce genre de chose, on peut traverser n\'importe quoi.

C: {ctx}
T: {en_text}
A:"""
    return templated_input
```

Pseudocode C.14 PaLM-with-Context **French Formality** Specialist Prompt Template

The PaLM-with Context *French Polysemy* specialist prompt template is the same for all test ambiguity data and is provided in code block listing C.15.

```
def french_baseline_polysemy_translator_context(en_text, ctx):
    """Translation model uses context to translate."""

    templated_input = f"""[web] Given context ’C’, Translate ’T’ from English to French:

C: Consequently a strategy has been defined that allows departments to approach its implementation in a step-wise manner.
T: approach
A: s’approcher, aborder, contacter, s’adresser

C: We need to abstract the data from various studies.
T: abstract
A: abstraire, extraire

C: About 2% of the households are enumerated using the canvasser method.
T: about
A: environ, presque, quelque, a peu pres, approximativement

C: The bat flew over the forest and back to its cave.
T: bat
A: chauve-souris

C: For the international community is not an abstract concept, it consists of us ourselves.
T: abstract
A: abstraction, abstrait

C: {ctx}
T: {en_text}
A:"""
    return templated_input
```

Pseudocode C.15 PaLM-with-Context **French Polysemy** Specialist Prompt Template

C.1.13 More details on gender and formality classifier

The classifiers fall into 2 categories: (1) heuristic based classification, that use the same language rules from section C.1.6; (2) neural network based classification, using a PaLM 62B model with 8-shot in-demonstration exemplars. We provide below the exemplars that were used to classify gender of French in code block listing C.16 and Spanish sentences in code block listing C.17. Note that we added exemplars until we had a satisfactory score on our ground truth translated sentence (see Table C.3).

```
def french_gender_it_classifier_template(en_text, fr_text):
  """Classify a French sentence as feminine or masculine. 7-shot examples"""

    templated_input =
f"""[web] Given French sentence 'F', provide the gender of "it" in English sentence 'T' and explain in 'E'. Gender in 'A' must be 'feminine', 'masculine' or 'neutral':


T: lily and marshall decided to sell it for one simple reason.
F: lyly et marshall l\'avaient mise en vente pour une seule raison.
A: feminine
E: It is 'feminine' since "mise" refers to a feminine object.


T: - maybe you need to shake it up.
F: - peut-etre qu'il faut le secouer.
A: masculine
E: It is 'masculine' since "le" refers to a masculine object.


T: i want you to get it for me.
F: Je veux que tu me la rapportes.
A: feminine
E: It is 'feminine' since "la" refers to a feminine object.


T: put it back.
F: repose-le.
```

```
A: masculine
E: It is ’masculine’ since "le" refers to a masculine object.


T: I’m afraid i won’t be able to get it for you.
F: Je crains de ne pas pouvoir te l’obtenir.
A: neutral
E: It is ’neutral’ since we cannot determine gender with "l\’" only.


T: that view is even more beautiful when you have someone to share it with.
F: elle est encore plus belle si on n’est pas seul.
A: feminine
E: It is ’feminine’ since "it" refers to "view" in English and "vue" in French which is feminine.


T: where’s it going?
F: ou va-t-il ?
A: masculine
E: It is ’masculine’ since "it" refers to "il" in French which is masculine.


T: {en_text}
F: {fr_text}
A: """
    return templated_input
```

Pseudocode C.16 PaLM prompt template for gender classification of French sentences

```
def spanish_gender_it_classifier_template(en_text, es_text):
    """Classify a Spanish sentence as feminine or masculine. 8-shot examples"""

    templated_input =

f"""[web] Given Spanish sentence ’F’, provide the gender in ’A’ and
```

```
   explain in ’E’. Gender ’A’ must be either ’feminine’ or ’
   masculine’:

F: nos habriamos pasado el dia mirandola.
A: feminine
E: It is ’feminine’ since "la" and verb "mirandola" refer to a
   feminine object.


F: - los peruanos no podian pronunciarlo.
A: masculine
E: It is ’masculine’ since "lo" in verb "pronunciarlo" refers to a
   masculine object.


F: Quiero decir, me encantaria volver a verlo.
A: masculine
E: It is ’masculine’ since "lo" in verb "verlo" refers to a
   masculine object.


F: debemos ponerla de vuelta?
A: feminine
E: It is ’feminine’ since "la" in verb "ponerla" refers to a
   feminine object.


F: -tiene que bebersela o tirarla.
A: feminine
E: It is ’feminine’ since "la" in verbs "bebersela" and "tirarla"
   refer to a feminine object.


F: Guardalo para el proximo barco.
A: masculine
E: It is ’masculine’ since "lo" in verb "Guardalo" refers to a
   masculine object.

```

```

F: \"escuchandola me dan ganas de vivir.\"
A: feminine
E: It is ’feminine’ since "la" in verb "escuchandola" refers to a feminine object.


F: !cambialo al menos!
A: masculine
E: It is ’masculine’ since "lo" in verb "cambialo" refers to a masculine object.


F: {es_text.lower()}
A: """
    return templated_input
```

Pseudocode C.17 PaLM prompt template for gender classification of Spanish sentences

We have added the classification heuristics and other classification templates to our public data and code repository.

Table C.3 PaLM 62B gender classification results on a 100 generated translation samples.

Spanish	**French**
97%	93%

APPENDIX D BLOCK STATE TRANSFORMERS

D.1 Appendix: Block State Transformers

D.1.1 Limitations

While BST's SSM layer allows the model to unroll and parallelize the recurrence that models long-term context between blocks of tokens, the SSM variants are reliant on efficient FFT operations. We have found that the FFT operation is an important speed bottleneck on TPUs that needs to be resolved to better scale BST to many layers and larger models. While we are still investigating the reasons, we found that JAX FFT was 4× faster on GPUs. Further, new SSM variants such as S5 [229] bypass FFT operations using a binary associative operator[1]. Our implementation is modular enough that we can simply plug in S5 or use other FFT implementations.

One of our assumptions is that BST's SSM layer is able to capture the right long-term dependencies for each block. The SSM recurrence at step $T = t$ provides a summarized representation of previous steps for $T = 0$ to $T = t$. However, a single vector representation may not be powerful enough to support all important long-term dependencies. Despite the perplexity improvements on long-range language modeling tasks, this assumption needs to be tested on other long-range classification tasks such as Long Range Arena [215] as well. It is possible that our model can perform better if we feed to the attention layer $k = W$ SSM representations that are chosen by a top-k retrieval operation, similar to the one in Memorizing Transformer [223].

D.1.2 More detailed comparisons with existing baselines

This section provides the reader with a more in-depth comparison with similar architectures. We cover BRecT [12] in Section D.1.2 and GSS-Hybrid [13] in Section D.1.2.

Comparison with Block Recurrent Transformer (BRecT)

The Block Transformer sublayer (i.e Slide:12L) processes keys and values from the previous window stored in a differentiable cache. This is implemented similarly to the sliding window attention pattern suggested in [12] and was originally introduced by Transformer-XL [208]. Using a causal mask, at every token inference step, the attention mechanism is applied to

[1] In JAX, this is equivalent to using *jax.lax.associative_scan*.

blocks of tokens of size W and is partially extended to the cached keys and values from the previous block with the sliding window. BRecT, as explained in [12], uses a non-differentiable cache that is carried from one sequence of size L to the next[2]. The last recurrent states of a sequence are stored in a non-differentiable cache and fed to the next training step on the following sequence in the document as a warm-start. We do not pass such a representation, since to compute the output of the convolution, we need access to the whole sequence. We believe that this is one advantage that BRecT has over our method, especially for very long examples that split into ordered sequences of length L, since the cache carried from one sequence to the next can provide very useful long-range information and (weak) access to the whole past. Since we need the whole sequence to compute SSM states, history beyond L may be lost in the process. We believe that BST can further be improved by adding non-differentiable sequence cache for very long documents.

While in other architectures, the history between blocks of tokens is not modeled, both BST and BRecT use a mechanism to model previous block context. The authors of BRecT experiment with various sequential gating mechanisms to condense the information from past blocks. With BST, we use SSM to provide context from previous blocks to the current block as explained in Section 5.3.2.

Comparison with the Transformer GSS-Hybrid

GSS-Hybrid [13] is a SSM-Transformer hybrid architecture that we first describe in Section 5.4.1. The architecture is significantly different from BST. GSS-Hybrid is primarily composed of Gated State Space (GSS) layers and has a few interleaved Transformer layers at every 4th layer starting with the 2nd layer. BST on the other hand is mainly composed of Block Transformer layers and has Block-State Transformer layers at positions $\{1, 7, 9\}$ for the $\sim$200M model and $\{1, 5, 7, 9\}$ for the $\sim$400M model. Our hybrid does not stack SSM and Transformer layers like the GSS-Hybrid but rather replaces the recurrence in BRecT with an SSM such as S4. In BST, the SSM generates states for each Block Transformer representations and we then use cross-attention to mix the states and the self-attention outputs. The authors in [13] initially built GSS, a gated version of DSS [231], to (1) reduce SSM parameter dimensions, (2) stabilize training of the SSM and (3) allow better length generalization. However, when experimenting with SSMs such as S4 or DSS, we found that the gating was not necessary to achieve all three objectives stated above. We decided that using GSS's Gated Attention Unit [227] was therefore not needed when integrating SSM states into

[2] In our work and in [12], a document is split into multiple sequences of size L and each sequence is split into multiple blocks of size W

the attention mechanism. We also reiterate that the authors in [13] used hyperparameter search to get the best performance while we did not.

D.1.3 JAX Implementation of BST

Pseudocode D.1 contains a function that implements convolution of multiple filters over the same input sequence using FFT and inverse FFT operations. Pseudocodes D.2, D.3 and D.4 respectively implement context state collection of BST variants: Single-Head (SH), Multi-Head (MH) and Multi-Filter (MF). Finally, Pseudocode D.5 runs the Block Transformer sublayer in parallel by feeding the context states to their corresponding block.

```python
"""Unstructured filters and convolutions."""

import jax
from jax import numpy as jnp
from einops import rearrange

win_length = 512    # (w)
seq_length = 4096   # (l)

def get_filters_unstruct(channels):
    """Returns trainable filters and biases.

    Args:
        channels: number of filters.

    Returns:
        h: filter of shape (seq_length, channels, dim)
        b: bias of shape (channels, dim)
    """
    t = jnp.linspace(0.0, 1.0, seq_length)
    h = jnp.exp(- alpha * t) * dense(positional_emb(t))
    b = get_bias()
    return h, b

def multichannel_convolution(u, h, b):
    """Multichannel convolution function.

    Args:
        u: input of shape (seq_length, dim)
        h: filters of shape (seq_length, channels, dim)
        b: bias of shape (channels, dim)
    """
    h = rearrange(h, "l c d -> c d l")

    fft_size = seq_length * 2
    u_f = jnp.fft.rfft(x, n=fft_size)
    h_f = jnp.fft.rfft(h, n=fft_size)

    y = jnp.fft.irfft(h_f * x_f, n=fft_size, norm="forward")[
            ..., :seq_length]       # (c, d, l)
    y = y + x * b[..., None]        # (c, d, l)
    y = rearrange(y, "c d l -> l d c")
    return y
```

Pseudocode D.1 Unstructured filters and convolutions.

```
"""Context state collection for BST-SH variant."""

num_heads = 8       # (h)
num_states = 32     # (s)

# (SH): Single-Head
def SH_context_states(u):
    """Single-Head Context Collection."""
    h, b = get_filters_[unstruct/s4](channels=1)
    y_1 = multichannel_convolution(u, h, b)  # y_1: (l, d, 1)

    # lift to multiple heads
    y_h = dense(y_1)  # y_h: (l, d, h)

    context_states = jnp.split(
            y_h, seq_length // win_length, axis=0)
    return context_states # (l/w, w, d, h)
```

Pseudocode D.2 Context state collection for BST-SH variants.

```
"""Context state collection for BST-MH variant."""

```

```
# (MH): Multi-Head
def MH_context_states(u):
    """Multi-Head Context Collection."""
    h, b = get_filters_[unstruct/s4](channels=num_heads)
    y_h = multichannel_convolution(u, h, b)  # y_h: (l, d, h)

    context_states = jnp.split(
            y_h, seq_length // win_length, axis=0)
    return context_states # (l/w, w, d, h)
```

Pseudocode D.3 Context state collection for BST-MH variants.

```
"""Context state collection for BST-MF variant."""

# (MF): Multi-Filter
def MF_context_states(u):
    """Multi-Filter Context Collection."""
    h, b = get_filters_[unstruct/s4](channels=num_states)
    y_s = multichannel_convolution(u, h, b)  # y_s: (l, d, s)
    context_states = jnp.split(
            y_s, seq_length // win_length, axis=0)
    # context_states: (l/w, w, d, s)

    # collect the last context states
    context_states = context_states[:, -1, ...] # (l/w, d, s)
    context_states = rearrange(
            context_states, "lw d s -> lw s d")

    # shift context states corresponding to windows
    context_states = jnp.roll(context_states, 1, axis=1)

    # replace the initial window with trainable weights
    init_context = get_init_context(num_states) # (d, s)
    context_states[0] = init_context

    # lift to multiple heads
    context_states = dense(context_states)

    return context_states # (l/w, s, d, h)
```

Pseudocode D.4 Context state collection for BST-MF variants.

```
"""Block-State Transformer Layer."""

# Block Transformers are non-recurrent and parallelizable.
block_transformer = jax.vmap(BRecT.nonrecurrent_cell)

def BST(u):
    """Block-State Transformer Layer."""
    global MF # True if Multi-Filter, False otherwise (SH/MH)

    # split inputs into windows (l/w, w, d)
    u = jnp.split(u, seq_length // win_length, axis=0)

    # collect context states from SSM outputs
    context_states = [SH/MH/MF]_context_states(u)

    # pass the contexts in place of recurrent states
    y = block_transformer(
            token_embeddings=u,
            recurrent_state=context_states,
            use_cross_attn_causal_mask=not MF,
            use_cross_positional_emb=MF, # context IDs
    )

    return rearrange(y, "lw w d -> (lw w) d") # (l, d)
```

Pseudocode D.5 Block-State Transformer Layer.