Dynamic Multi-Level Multi-Task Learning for Sentence Simplification

Han Guo **Ramakanth Pasunuru** **Mohit Bansal**
UNC Chapel Hill
{hanguo, ram, mbansal}@cs.unc.edu

Abstract

Sentence simplification aims to improve readability and understandability, based on several operations such as splitting, deletion, and paraphrasing. However, a valid simplified sentence should also be logically entailed by its input sentence. In this work, we first present a strong pointer-copy mechanism based sequence-to-sequence sentence simplification model, and then improve its entailment and paraphrasing capabilities via multi-task learning with related auxiliary tasks of entailment and paraphrase generation. Moreover, we propose a novel 'multi-level' layered soft sharing approach where each auxiliary task shares different (higher versus lower) level layers of the sentence simplification model, depending on the task's semantic versus lexico-syntactic nature. We also introduce a novel multi-armed bandit based training approach that dynamically learns how to effectively switch across tasks during multi-task learning. Experiments on multiple popular datasets demonstrate that our model outperforms competitive simplification systems in SARI and FKGL automatic metrics, and human evaluation. Further, we present several ablation analyses on alternative layer sharing methods, soft versus hard sharing, dynamic multi-armed bandit sampling approaches, and our model's learned entailment and paraphrasing skills.

1 Introduction

Sentence simplification is the task of improving the readability and understandability of an input text. This challenging task has been the subject of research interest because it can address automatic ways of improving reading aids for people with limited language skills, or language impairments such as dyslexia (Rello et al., 2013), autism (Evans et al., 2014), and aphasia (Carroll et al., 1999). It also has wide applications in NLP tasks as a preprocessing step, for example, to improve the performance of parsers (Chandrasekar et al., 1996), summarizers (Klebanov et al., 2004), and semantic role labelers (Vickrey and Koller, 2008; Woodsend and Lapata, 2014).

Several sentence simplification systems focus on operations such as splitting a long sentence into shorter sentences (Siddharthan, 2006; Petersen and Ostendorf, 2007), deletion of less important words/phrases (Knight and Marcu, 2002; Clarke and Lapata, 2006; Filippova and Strube, 2008), and paraphrasing (Devlin, 1999; Inui et al., 2003; Kaji et al., 2002). Inspired from machine translation based neural models, recent work has built end-to-end sentence simplification models along with attention mechanism, and further improved it with reinforcement-based policy gradient approaches (Zhang and Lapata, 2017). Our baseline is a novel application of the pointer-copy mechanism (See et al., 2017) for the sentence simplification task, which allows the model to directly copy words and phrases from the input to the output. We further improve this strong baseline by bringing in auxiliary entailment and paraphrasing knowledge via soft and dynamic multi-level, multi-task learning.[1]

Apart from the three simplification operations discussed above, we also ensure that the simplified output is a directed logical entailment w.r.t. the input text, i.e., does not generate any contradictory or unrelated information. We incorporate this entailment skill via multi-task learning (Luong et al., 2015)

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

[1] All code and pretrained models available at: https://github.com/HanGuo97/MultitaskSimplification.

Proceedings of the 27th International Conference on Computational Linguistics, pages 462–476
Santa Fe, New Mexico, USA, August 20-26, 2018.

with an auxiliary entailment generation task. Further, we also induce word/phrase-level paraphrasing knowledge via a paraphrase generation task, enabling parallel learning of these three tasks in a three-way multi-task learning setup. We employ a novel 'multi-level' layered, soft sharing approach, where the parameters between the tasks are loosely coupled at different levels of layers; we share higher-level semantic layers between the sentence simplification and entailment generation tasks (which teaches the model to generate outputs that are entailed by the full input), while sharing the lower-level lexico-syntactic layers between the sentence simplification and paraphrase generation tasks (which teaches the model to paraphrase only the smaller sub-sentence pieces).

Finally, we also propose a multi-armed bandit approach that dynamically learns an effective schedule (curriculum) of switching between tasks for optimization during multi-task learning, instead of the traditional approach with a manually-tuned static (fixed) mixing ratio (Luong et al., 2015).

Empirically, we evaluate our system on three standard datasets: Newsela, WikiSmall, and WikiLarge. First, we show that our pointer-copy baseline is significantly better than sequence-to-sequence models, and competitive w.r.t. the state-of-the-art. Next, we show that our multi-level, multi-task framework performs significantly better than our strong pointer baseline and other competitive sentence simplification models on both automatic evaluation as well as on human study simplicity criterion. Further, we show that the dynamic multi-armed bandit based switching of tasks during training improves over the traditional manually-tuned static mixing ratio. Lastly, we show several ablation studies based on different layer-sharing approaches (higher versus lower) with auxiliary tasks, hard versus soft sharing, dynamic mixing ratio sampling, as well as our model's learned entailment and paraphrasing skills.

2 Related Work

Previous approaches to sentence simplification systems range from hand-designed rules (Siddharthan, 2006), to syntactic and lexical simplification via synonyms and paraphrases (Siddharthan, 2014; Kaji et al., 2002; Horn et al., 2014; Glavaš and Štajner, 2015), as well as treating simplification as a monolingual MT task, where operations are learned from examples of complex-simple sentence pairs (Specia, 2010; Koehn et al., 2007; Coster and Kauchak, 2011; Zhu et al., 2010; Wubben et al., 2012; Narayan and Gardent, 2014). Recently, Xu et al. (2016) trained a syntax-based MT model using the newly proposed SARI as a simplification-specific objective. Further, Zhang and Lapata (2017) used reinforcement learning in a sequence-to-sequence approach to directly optimize simplification metrics. In this work, we first introduce the pointer-copy mechanism (See et al., 2017) as a novel application to sentence simplification, and then use multi-task learning to bring in auxiliary entailment and paraphrasing skills.

Multi-task learning, known for improving the generalization performance of a task with related tasks, has successful application to many domains of machine learning (Caruana, 1998; Collobert and Weston, 2008; Girshick, 2015; Luong et al., 2015; Pasunuru and Bansal, 2017; Pasunuru et al., 2017). Although there are many variants of multi-task learning (Ruder et al., 2017; Hashimoto et al., 2017; Luong et al., 2015), our approach is similar to Luong et al. (2015), where different tasks share some common model parameters with alternating mini-batches optimization. In this work, we explore a multi-level (i.e., task-specific higher-level semantic versus lower-level lexico-syntactic layer sharing) and soft-sharing mechanism for improving sentence simplification via related tasks of entailment and paraphrase generation.

Recognizing Textual Entailment (RTE) is the task of predicting entailment, contradiction, or neutral relationships, and is useful for many downstream tasks like Q&A, summarization, and information retrieval (Harabagiu and Hickl, 2006; Dagan et al., 2006; Lai and Hockenmaier, 2014; Jimenez et al., 2014). Neural network models (Bowman et al., 2015; Parikh et al., 2016) and large datasets (Bowman et al., 2015; Williams et al., 2017) enabled recent strong progress. Recently, Pasunuru et al. (2017) and Guo et al. (2018) presented results using entailment generation as an auxiliary task for abstractive summarization; however, we use entailment as well as paraphrasing knowledge in a soft and multi-level layer sharing setup to improve sentence simplification.

Previous work (Barzilay and McKeown, 2001; Ganitkevitch et al., 2013; Wieting and Gimpel, 2017a) has developed methods and datasets for generating paraphrase pairs which can be useful for downstream applications such as question answering, semantic parsing, and information extraction (Fader et al., 2013;

Berant and Liang, 2014; Zhang et al., 2015). Wieting and Gimpel (2017a) recently introduced a large sentential paraphrase dataset via back-translation, and showed promising results when applied to learning sentence embeddings. In this work, we use this paraphrase dataset as an auxiliary generation task to improve our sentence simplification model by teaching it about paraphrasing in a multi-task setting.

Many control problems can be cast as a multi-armed bandits algorithm, where the goal of the agent is to select the arm/action from one of the M choices that gives the maximum expected future reward (Bubeck et al., 2012). Optimal control and reinforcement learning have been used to find the trade-off between exploitation and exploration, and yield theoretically-sound regret bounds, e.g., Boltzmann exploration (Kaelbling et al., 1996), UCB (Auer et al., 2002a), Thompson sampling (Chapelle and Li, 2011), adversarial bandits (Auer et al., 2002b), and information gain using variational approaches (Houthooft et al., 2016). Recently, Graves et al. (2017) use a non-stationary multi-armed bandit to automatically select the curriculum or syllabus that a neural network follows so as to maximize learning efficiency. Sharma and Ravindran (2017) use multi-armed bandit sampling to choose which domain data (harder vs. easier) to feed as input to a single model (using different Atari games), whereas we use multi-armed bandit sampling to decide the optimization curriculum (mixing ratio) among our three models for sentence simplification, entailment generation, and paraphrase generation (with different softly-shared layers).

3 Models

In this section, we first describe our sentence simplification baseline model with attention mechanism, which is further improved by pointer-copy mechanism. Later, we introduce our two auxiliary tasks (entailment and paraphrase generation) and discuss how they can share specific lower/higher-level layers/parameters to improve the sentence simplification task in a multi-task learning setting. Finally, we discuss our new multi-armed bandit based dynamic multi-task learning approach.

3.1 Pointer-Copy Baseline Sentence Simplification Model

Our baseline is a 2-layer sequence-to-sequence model with both attention (Bahdanau et al., 2015) and pointer-copy mechanism (See et al., 2017). Given the sequence of input/source tokens $x = \{x_1, ..., x_{T_s}\}$, the model learns an auto-regressive distribution over output/target tokens $y = \{y_1, ..., y_{T_o}\}$, which is defined as $P_{vocab}(y|x;\theta) = \prod_t p(y_t|y_{1:t-1}, x;\theta)$, where θ represents model parameters and $p(y_t|y_{1:t-1}, x;\theta)$ is probability of generating token y_t at decoder time step t given the previous generated tokens $y_{1:t-1}$ and input x. Given encoder hidden states $\{h_i\}$, and decoder's t^{th} time step hidden state (of last layer) s_t, the context vector $c_t = \sum_i \alpha_{t,i} h_i$, where the attention weights $\alpha_{t,i}$ define an attention distribution over encoder hidden states: $\alpha_{t,i} = \exp(e_{t,i}) / \sum_k \exp(e_{t,k})$, where $e_{t,i} = v_a^T \tanh(W_a s_t + U_a h_i + b_a)$. Finally, the conditional distribution at each time step t of the decoder is defined as $p(y_t|y_{1:t-1}, x;\theta) = \text{softmax}(W_s s'_t)$, where the final hidden state s'_t is a combination of context vector c_t and last layer hidden state s_t and is defined as $s'_t = \tanh(W_c[c_t, s_t])$, where W_s and W_c are trained parameters.

Pointer-Copy Mechanism: This helps in directly copying the words from the source inputs to the target outputs via merging the generative distribution and attention distribution (as a proxy of copy distribution). The goal of sentence simplification is to rewrite sentences more simply, while preserving important information; hence, it also involves significant amount of copying from the source. Our pointer mechanism approach is similar to See et al. (2017). At each time step of the decoder, the model makes a (soft) choice between words from the vocabulary distribution P_{vocab} and attention distribution P_{att} (based on words in the input) using the word generation probability $p_g = \sigma(W_g c_t + U_g s_t + V_g d_t + b_g)$, where $\sigma(\cdot)$ is sigmoid, W_g, U_g, V_g and b_g are trainable parameters, and d_t is decoder input. The final vocabulary distribution is defined as the weighted combination of vocabulary and attention distributions:

$$P_f(y) = p_g P_{vocab}(y) + (1 - p_g) P_{att}(y) \tag{1}$$

3.2 Auxiliary Tasks

Entailment Generation The task of entailment generation is to generate a hypothesis which is entailed by the given input premise. A good simplified sentence should be entailed by (follow from) the source

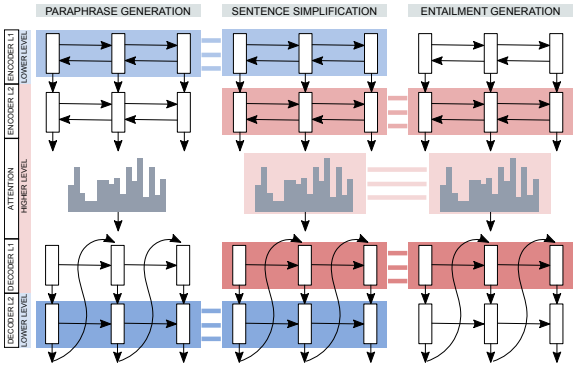

Figure 1: Overview of our 3-way multi-task model. Same color and dashed connections represent soft-shared parameters in different layers.

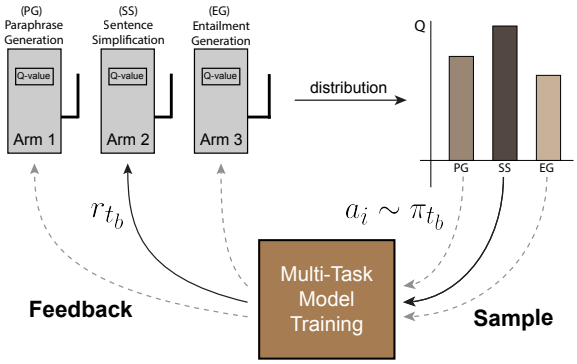

Figure 2: Overview of our multi-armed bandits algorithm for dynamic mixing ratio learning. It consists of a controller with 3 arms/tasks.

sentence, and hence we incorporate such knowledge through an entailment generation task into our sentence simplification task. We share the higher-level semantic layers between the two tasks (see reasoning in Sec. 3.3 below). We use entailment pairs from SNLI (Bowman et al., 2015) and Multi-NLI (Williams et al., 2017) datasets for training our entailment generation model, where we use the same architecture as our sentence simplification model.

Paraphrase Generation Paraphrase generation is the task of generating similar meaning phrases or sentences by reordering and modifying the syntax and/or lexicon. Paraphrasing is one of the common operations used in sentence simplification, i.e, by substituting complex words and phrases with their simpler paraphrase forms. Hence, we add this knowledge to the sentence simplification task via multi-task learning, by sharing the lower-level lexico-syntactic layers between the two tasks (see reasoning in Sec. 3.3 below). For this, we use the paraphrase pairs from ParaNMT (Wieting and Gimpel, 2017a). Here, again, we use the same architecture as our sentence simplification model.

3.3 Multi-Task Learning

In this subsection, we discuss our multi-task, multi-level soft sharing strategy with parallel training of sentence simplification and related auxiliary tasks (entailment and paraphrase generation).

The predominant approach for multi-task learning in sequence-to-sequence models is to directly hard-share all encoder/decoder layers/parameters (Luong et al., 2015; Johnson et al., 2016; Pasunuru and Bansal, 2017; Kaiser et al., 2017). However, this approach places very strong constraints/priors on the primary model to compress knowledge from diverse tasks. We believe that while the auxiliary tasks considered in this work share many similarities with the primary sentence simplification task, they are still different in either lower-level or higher-level representations (e.g., entailment will deal with higher-level, full-sentence logical inference, while paraphrasing will handle the lower-level intermediate word/phrase simplifications). In this section, we propose to relax the priors in two ways: (1) we share the model parameters in a finer-grained scale, i.e. layer-specific sharing, by keeping some of their parameters private, while sharing related representations; and (2) we encourage shared parameters to be close in certain distance metrics with a penalty term instead of hard-parameter-tying (Luong et al., 2015).

Multi-Level Sharing Mechanism Fig. 1 shows our multi-task model with parallel training of three tasks: sentence simplification (primary task), entailment generation (auxiliary task), and paraphrase generation (auxiliary task). Recently, Belinkov et al. (2017) observed that different layers in a sequence-to-sequence model (trained on translation) exhibit different functionalities: lower-layers (closer to inputs) of the encoder learn to represent word structure while higher layers (farther from inputs) are more focused on semantics and meanings (Zeiler and Fergus (2014) observed similar findings for convolutional image features). Based on these findings, we share the higher-level layers[2] between the entailment generation and sentence simplification tasks, since they share higher semantic-level language inference

[2]We found that sharing higher-level semantic layers (farther from input/output), i.e., encoder layer 2, attention, and decoder layer 1 (in Fig. 1), to work well. See Sec. 6 for ablations on alternative layer sharing methods.

skills (for full sentence-to-sentence logical directedness). On the other hand, we share the lower-level lexico-syntactic layers[3] between the paraphrase generation and sentence simplification tasks, since they share more word/phrase and syntactic level paraphrasing knowledge to simplify the smaller, intermediate sentence pieces. Sec. 6 present empirical ablations to support our intuitive layer sharing.[4]

Soft Sharing In multi-task learning, we can do either hard sharing or soft sharing of parameters. Hard sharing directly ties the parameters to be shared, and receives gradient information from multiple tasks. On the other hand, soft sharing only loosely couples the parameters, and encourages them to be close in representation space. Hence the soft sharing approach gives more flexibility for parameter sharing, hence allowing different tasks to choose what parts of their parameters space to share. We minimize the l_2 distance between shared parameters as a regularization along with the cross entropy loss. Hence, the final loss function of the primary task with a related auxiliary task is defined as follows:

$$L(\theta) = -\log P_f(y|x;\theta) + \lambda||\theta_s - \phi_s|| \tag{2}$$

where θ represents the full parameters of the primary task (sentence simplification), θ_s and ϕ_s are the subsets of shared parameters between the primary and auxiliary task resp., and λ is a hyperparameter.

Multi-Task Training We employ multi-task learning with parallel training of related tasks in alternate mini-batches based on a mixing ratio α_{ss}:α_{eg}:α_{pp}, where we alternatively optimize α_{ss}, α_{eg}, α_{pp} mini-batches of sentence simplification, entailment generation, and paraphrase generation, respectively, until all models converge. In the next section, we discuss a new approach to replace this static mixing ratio with dynamically-learned task switching.

3.4 Dynamic Mixing Ratio Learning

Current multi-task models are trained via alternate mini-batch optimization based on a task 'mixing ratio' (Luong et al., 2015; Pasunuru and Bansal, 2017), i.e., how many iterations on each task relative to other tasks (see end of Sec. 3.3). This is usually treated as a very important hyperparameter to be tuned, and the search space scales exponentially with the number of tasks. Hence, we importantly replace this manually-tuned and static mixing ratio with a 'dynamic' mixing ratio learning approach, where a controller automatically switches between the tasks during training, based on the current state of the multi-task model. Specifically, we use a multi-armed bandits based controller with Boltzmann exploration (Kaelbling et al., 1996) with an exponential moving average update rule.

We view the problem of learning the right mixing of tasks as a sequential control problem, where the controller's goal is to decide the next task/action after every n^s training steps in each task-sampling round t_b.[5] Let $\{a_1, ..., a_M\}$ represent the set of 3 tasks in our multi-task setting, i.e., sentence simplification, entailment generation, and paraphrase generation. We model the controller as a M-armed bandits, where it selects a sequence of actions/arms over the current training trajectory to maximize the expected future payoffs (see Fig. 2). At each round t_b, the controller selects an arm based on noisy value estimates and observes rewards r_{t_b} for the selected arm (we use the negative validation loss of the primary task as the reward in our setup). One problem in bandits learning is the trade-off between exploration and exploitation, where the agent needs to make a decision between taking the action that yields the best payoff on current estimates, or explore new actions whose payoffs are not yet certain. For this, we use the Boltzmann exploration (Kaelbling et al., 1996) with exponentially moving action value estimates. Let π_{t_b} be the policy of the bandit controller at round t_b, we define this to be:

$$\pi_{t_b}(a_i) = \exp(Q_{t_b,i}/\tau) \Big/ \sum_{j=1}^{M} \exp(Q_{t_b,j}/\tau) \tag{3}$$

[3]We found that sharing lower-level lexico-syntactic layers (closer to input/output), i.e., encoder layer 1 and decoder layer 2 (in Fig. 1), to work well. See Sec. 6 for ablations on alternative layer sharing methods.

[4]Note that even though entailment just tries to generate shorter, logical-subset sub-sentences, the overall saliency and quality of the simplified output is still balanced because the entailment task is flexibly (softly) shared with the paraphrasing and sentence simplification tasks, and the final model mixture is chosen based on simplification task metrics (see output examples in Fig. 4 where our multi-task model generates entailed sentences with important information).

[5]We set n^s to 10 to reduce variance of estimates, i.e., the bandit controller's task/action will be trained for 10 mini-batches.

where $Q_{t_b,i}$ is the estimated action value of each arm i at round t_b, and τ is the temperature.[6] If $Q_{0,i}$ is the initial value estimate of arm i, then $Q_{t_b,i}$ is the exponentially weighted mean with the decay rate α:

$$Q_{t_b,i} = (1-\alpha)^{t_b} Q_{0,i} + \sum_{k=1}^{t_b} \alpha(1-\alpha)^{t_b-k} r_k \tag{4}$$

To further help the exploration process, we follow the principle of optimism under uncertainty (Sutton and Barto, 1998) and set $Q_{0,i}$ to be above the maximum empirical rewards. Empirically, we show that this approach of 'dynamic mixing ratio' is equal or better than the traditional static mixing ratio (see Table 3). Also, we further show ablation study in Sec. 6 to show that this switching approach is better than the alternative approach of first using multi-armed bandits for finding an optimal 'final' mixing ratio and then re-training the model based on this bandits-selected mixing ratio.

4 Evaluation Setup

Datasets We first describe the three standard sentence simplification datasets we evaluate on: Newsela, WikiSmall, and WikiLarge; next, we describe datasets for our auxiliary entailment and paraphrase generation tasks. **Newsela** (Xu et al., 2015) is acknowledged as a higher-quality dataset for studying sentence simplifications, as opposed to Wikipedia-based datasets which automatically align complex-simple sentence pairs and have generalization issues (Zhang and Lapata, 2017; Xu et al., 2015; Amancio and Specia, 2014; Hwang et al., 2015; Štajner et al., 2015). Newsela consists of $1,130$ news articles, and we follow previous work (Zhang and Lapata, 2017) to use the first $1,070$ documents for training, and 30 documents each for development and test. **WikiSmall** (Zhu et al., 2010) contains automatically-aligned complex-simple sentences from the ordinary-simple English Wikipedias. The data has $89,042$ pairs for training and 100 for test. We use the 205-pairs validation set from Zhang and Lapata (2017). **WikiLarge** (Zhang and Lapata, 2017) is a larger Wikipedia corpus aggregating pairs from Kauchak (2013), Woodsend and Lapata (2011), and WikiSmall. We use the exact training/evaluation sets provided by Zhang and Lapata (2017). **SNLI and MultiNLI**: For the task of entailment generation, we use the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) and MultiNLI (Williams et al., 2017). We use their entailment labeled pairs for our entailment generation task, following previous work (Pasunuru and Bansal, 2017). The combined SNLI and MultiNLI dataset has $302,879$ entailment pairs, out of which we use $276,720$ pairs for training, and the rest are divided into validation and test sets. **ParaNMT**: For the task of paraphrase generation, we use the back-translated paraphrase dataset provided by Wieting and Gimpel (2017a). The filtered version of the dataset has 5.3 million pairs of paraphrases.[7] We use 99% for training, and the rest are evenly divided into validation and test sets.

Evaluation Metrics Following previous work (Zhang and Lapata, 2017), we report all the standard evaluation metrics: SARI (Xu et al., 2016), FKGL (Kincaid et al., 1975), and BLEU (Papineni et al., 2002). However, several studies have shown that BLEU is poorly correlated w.r.t. simplicity (Zhu et al., 2010; Štajner et al., 2015; Xu et al., 2016). Moreover, Shardlow (2014) argues that FKGL (Kincaid et al., 1975), which measures readability of simpler output (lower is better), favors very short sentences even though longer/less coarse counterparts can be simpler. Further, Xu et al. (2016) argues that BLEU tends to favor conservative systems that do not make many changes, and proposes SARI metric which explicitly measures the quality of words that are added and deleted. SARI is shown to correlate well with human judgment in simplicity (Xu et al., 2016), and hence we primarily focus on this metric in our models' performance analysis.[8] Further, we also do human evaluation based on: Fluency ('is the output grammatical and well formed?'), Adequacy ('to what extent is the meaning expressed in the original sentence preserved in the output?') and Simplicity ('is the output simpler than the original sentence?'), following guidelines suggested by Xu et al. (2016) and Zhang and Lapata (2017).

[6] We tried decaying the temperature variable, but we didn't find this to very beneficial, so we instead fix this to 1.0.

[7] We chose ParaNMT over other paraphrase datasets (e.g. the phrase-to-phrase PPDB dataset (Ganitkevitch et al., 2013)), because ParaNMT is a sentence-to-sentence dataset and hence is a more natural fit for sentence-level multi-task RNN-layer sharing with our sentence-to-sentence simplification task.

[8] We use the JOSHUA package for calculating SARI and BLEU score following Zhang and Lapata (2017) and Xu et al. (2016). Our FKGL implementation is based on `https://github.com/mmautner/readability`.

Models	BLEU	FKGL	SARI
PREVIOUS WORK			
PBMT-R	18.19	7.59	15.77
Hybrid	14.46	4.01	30.00
EncDecA	21.70	5.11	24.12
DRESS	23.21	4.13	27.37
DRESS-LS	24.30	4.21	26.63
OUR MODELS			
Baseline ⊗	23.72	3.25	28.31
⊗ + Ent.	16.82	2.21	31.55
⊗ + Paraphr.	16.29	2.03	31.71
⊗+Ent+Par	11.86	1.38	32.98

Table 1: Newsela (FKGL: lower is better).

	WIKISMALL			WIKILARGE		
Models	BLEU	FKGL	SARI	BLEU	FKGL	SARI
PREVIOUS WORK						
PBMT-R	46.31	11.42	15.97	81.11	8.33	38.56
Hybrid	53.94	9.21	30.46	48.97	4.56	31.40
SBMT-SARI	-	-	-	73.08	7.29	39.96
EncDecA	47.93	11.35	13.61	88.85	8.41	35.66
DRESS	34.53	7.48	27.48	77.18	6.58	37.08
DRESS-LS	36.32	7.55	27.24	80.12	6.62	37.27
OUR MODELS						
Baseline ⊗	36.18	7.69	25.67	82.37	7.84	36.68
⊗+Ent+Par	29.70	6.93	28.24	81.49	7.41	37.45

Table 2: WikiSmall/Large results (FKGL: lower is better).

Training Details All our soft/hard and layer-specific sharing decisions (Sec. 6) were made on the validation/dev set. Our model selection (tuning) criteria is based on the average of our 3 metrics (SARI, BLEU, 1/FKGL) on the validation set. Please refer to the appendix for full training details (vocabulary overlap, mixing ratios and bandit sampler decay rates and reward, WikiLarge pre-training, etc.).

5 Results

We evaluate our models on three datasets and via several automatic metrics plus human evaluation.[9]

Pointer Baseline First, we compare our pointer baseline with various previous works: PBMT-R (Wubben et al., 2012), Hybrid (Narayan and Gardent, 2014), SBMT-SARI (Xu et al., 2016)[10], and EncDecA, DRESS, and DRESS-LS (Zhang and Lapata, 2017). As shown in Table 1, our pointer baseline already achieves the best score in FKGL and the second-best score in SARI on Newsela, and also achieves overall comparable results on both WikiSmall and WikiLarge (see Table 2).

Multi-Task Models We further improve our strong pointer-based sentence simplification baseline model by multi-task learning it with entailment and paraphrase generation. First, we show that our 2-way multi-task models with auxiliary tasks (entailment and paraphrase generation) are statistically significantly better than our pointer baseline and previous works in both SARI and FKGL on Newsela (see Table 1).[11] Next, Table 1 and Table 2 summarize the performance of our final 3-way multi-level, multi-task models with entailment generation and paraphrase generation on all three datasets. Here, our 3-way multi-task models are statistically significantly better than our pointer baselines in both SARI and FKGL (with $p < 0.01$) on Newsela and WikiSmall, and in SARI ($p < 0.01$) on WikiLarge. Also, our 3-way multi-task model is statistically significantly better than the 2-way multi-task models in SARI and FKGL with $p < 0.01$ (see Table 1). In Sec. 6, we further provide a set of detailed ablation experiments investigating the effects of different (higher-level versus lower-level) layer sharing methods and soft- vs. hard-sharing in our multi-level, multi-task models; and we show the superiority of our final choice of higher-level semantic sharing for entailment generation and lower-level lexico-syntactic sharing for paraphrase generation.

Models	BLEU	FKGL	SARI
NEWSELA			
Static Mixing Ratio	11.86	1.38	32.98
Dynamic Mixing Ratio	11.14	1.32	33.22
WIKISMALL			
Static Mixing Ratio	29.70	6.93	28.24
Dynamic Mixing Ratio	27.23	5.86	29.58

Table 3: Results on dynamic vs. static mixing ratio (FKGL: lower is better).

Dynamic Mixing Ratio Models Finally, we present results on our 3-way multi-task model with the new approach of using 'dynamic' mixing ratios based on multi-armed bandits sampling (see Sec. 3.4).

[9] As described in Sec. 4, Newsela is considered as a higher quality dataset for text simplification, and thus we report ablation-style results (e.g., 2-way multi-task models and different layer-sharing ablations) and human evaluation on Newsela (since Wikipedia datasets are automatically-aligned). Moreover, we report SARI, FKGL, and BLEU for completeness, but as described in Sec. 4, SARI is the primary human-correlated metric for sentence simplification.

[10] We borrow the SBMT-SARI results for WikiLarge from Zhang and Lapata (2017).

[11] Stat. significance is computed via bootstrap test (Noreen, 1989; Efron and Tibshirani, 1994). Both our 2-way multi-task models are statistically significantly better in SARI and FKGL with $p < 0.01$ w.r.t. our pointer baseline and previous works. Note the discussion in Sec. 4 about why BLEU is not a good sentence simplification metric.

Models	HUMAN EVALUATION				MATCH WITH INPUT		
	Fluency	Adequacy	Simplicity	Average	BLEU (%)	ROUGE (%)	Exact Match (%)
Ground-truth	4.97	4.08	3.83	4.29	18.25	43.74	0.00
Hybrid	3.88	3.82	3.92	3.87	25.74	56.20	3.34
DRESS-LS	4.84	4.18	3.21	4.08	42.93	67.61	14.48
Pointer Baseline	4.61	3.94	3.99	4.18	30.80	60.56	10.68
3-way Multi-task	4.73	3.18	4.62	4.18	8.74	37.82	2.41

Table 4: Human evaluation results (on left) and closeness-to-input source results (on right), for Newsela.

As shown in Table 3, this dynamic multi-task approach achieves a stat. significant improvement in SARI as compared to the traditional fixed and manually-tuned mixing ratio based 3-way multi-task model: 33.22 vs. 32.98 ($p < 0.05$) on Newsela, and 29.58 vs. 28.24 ($p < 0.001$) on WikiSmall. Hence, this allows us to achieve better results while also avoiding the hassle of tuning on the large space of mixing ratios over several different tasks. In Sec. 6, we further provide ablation analysis to study whether the improvements come from the bandit learning this dynamic curriculum or from the bandit finding the final optimal mixing-ratio at the end of the sampling procedure (and also compare it to a random curriculum).

Human Evaluation We also perform an anonymized human study comparing our pointer baseline, our multi-task model, some previous works (Hybrid (Narayan and Gardent, 2014) and state-of-the-art DRESS-LS (Zhang and Lapata, 2017)), and ground-truth references (see left part of Table 4), based on fluency, adequacy, and simplicity (see Sec. 4 for more details about these criteria) using 5-point Likert scale. We asked annotators to evaluate the models (randomly shuffled to anonymize model identity) based on 200 samples from the representative and cleaner Newsela test set, and their scores are reported in Table 4. Our 3-way multi-task model achieves a significantly higher ($p < 0.001$) simplicity score compared to DRESS-LS, Hybrid, and our pointer baseline models. However, we next observe that our 3-way multi-task model has lower adequacy score as compared to DRESS-LS and the pointer model, but this is because our 3-way multi-task model focuses more strongly on simplification, which is the goal of the given task. Moreover, based on the overall average score of the three human evaluation criteria, our 3-way multi-task model is also significantly better ($p < 0.03$) than the state-of-the-art DRESS-LS model (and $p < 0.001$ w.r.t. Hybrid model).[12] Also, on further investigation, we found that a problem with the adequacy metric is that it gets artificially high scores for output sentences which are exact match (or a very close match) with the input source sentence, i.e., they have very little simplification and hence almost fully retain the exact meaning. In the right part of Table 4, we analyzed the matching scores of the outputs from different models w.r.t. the source input text, based on BLEU, ROUGE (Lin, 2004) and exact match. First, this shows that the ground-truth sentence-simplification references are in fact (as expected) very different from the input source (0% exact match, 18% BLEU, 44% ROUGE). Next, we find that our multi-task model also has low match-with-input scores (2% exact match, 9% BLEU, 38% ROUGE), similar to the behavior of the ground-truth references. On the other hand, DRESS-LS (and pointer baseline) model is generating output sentences which are substantially closer to the input and hence is not making enough changes (14% exact match, 43% BLEU, 68% ROUGE) as compared to the references (which explains their higher adequacy but lower simplicity scores).

6 Ablations and Analysis

In this section, we conduct several ablation analyses to study the different layer-sharing mechanisms (higher semantic vs. lower lexico-syntactic), soft- vs. hard-sharing, two dynamic multi-armed bandit approaches, and our model's learned entailment and paraphrasing skills. We also present and analyze some output examples from several models.[13] Note that all our soft and layer sharing decisions were strictly made on the dev/validation set (see Sec. 4).

[12]Note that our multi-task model is stat. equal to our pointer baseline on the overall-average score, showing the available trade-off between systems that simplify conservatively vs. strongly, based on one's desired downstream task application. Also refer to the high 'match-with-input' issue with the adequacy metric discussed next.

[13]Since Newsela is considered as the more representative dataset for sentence simplification with lesser noise and human quality (Xu et al., 2015; Zhang and Lapata, 2017), we conduct our ablation studies on this dataset, but we observed similar patterns on the other two datasets as well.

Models	Entailment	Paraphrasing
Ground-truth	N/A	62.1
Hybrid	34.8	74.1
DRESS-LS	30.7	77.9
Pointer Baseline	36.9	76.6
2-way Multi-Task	41.4	63.9

Table 6: Analysis: Entailment and paraphrase classification results (avg. probability scores as %) on Newsela.

Models	Deletions	Additions
Hybrid	95.18	0.000
DRESS-LS	85.37	0.047
Pointer Baseline	88.91	0.026
3-way Multi-Task	97.54	0.049

Table 7: Analysis: SARI's sub-operation scores on Newsela dataset.

Models	BLEU	FKGL	SARI
Final (High Ent + Low PP)	**11.86**	**1.38**	**32.98**
Both lower-layer	11.94	1.47	31.92
Both higher-layer	12.26	1.38	32.02
Swapped (Low Ent + High PP)	21.64	2.97	29.07
Hard-sharing	13.01	1.38	32.36

Table 5: Multi-task layer ablation results on Newsela.

Different Layer Sharing Approaches We empirically show that our final multi-level layer sharing method (i.e., higher-level semantic layer sharing with entailment generation, while lower-level lexico-syntactic layer sharing with paraphrase generation) performs better than the following alternative layer sharing methods: (1) both auxiliary tasks with high-level layer sharing, (2) both with low-level layer sharing, and (3) reverse/swapped sharing (i.e. lower-level layer sharing for entailment, and higher-level layer sharing for paraphrasing). Results in Table 5 show that our approach of high-level sharing for entailment generation and low-level sharing for paraphrase generation is statistically significantly better than all other alternative approaches in SARI ($p < 0.01$) (and statistically better or equal in FKGL).

Soft- vs. Hard-Sharing In this work, we use soft-sharing instead of hard-sharing approach (benefits discussed in Sec. 3.3) in all of our models. Table 5 also presents empirical results comparing soft- vs. hard-sharing on our final 3-way multi-task model, and we observe that soft-sharing is statistically significantly better than hard-sharing in SARI with $p < 0.01$.

Quantitative Improvements in Entailment We employ a state-of-the-art entailment classifier (Chen et al., 2017) to calculate the entailment probabilities of output sentence being entailed by the ground-truth.[14] Table 6 summaries the average entailment scores for the Hybrid, DRESS-LS, Pointer baseline, and 2-way multi-task model (with entailment generation auxiliary task), showing that the 2-way multi-task model improves in the aspect of logical entailment ($p < 0.001$), demonstrating the inference skill acquired by the simplification model via the auxiliary knowledge from the entailment generation task.

Quantitative Improvements in Paraphrasing We use the paraphrase classifier from Wieting and Gimpel (2017b) to compute the paraphrase probability score between the generated output and the input source. The results in Table 6 show that our 2-way multi-task model (with paraphrasing generation auxiliary task) is closer to the ground-truth in terms of the amount of paraphrasing (w.r.t. input) required by the sentence-simplification task, while the pointer baseline and previous models have higher scores due to higher amount of copying from input source (see 'Match-with-Input' discussion in Sec. 5, Table 4).

Addition/Deletion Operations We also measured the performance of the various models in terms of the addition and deletion operations using SARI's sub-operation scores computed w.r.t. both the ground-truth and source (Xu et al., 2016). Table 7 shows that our multi-task model is equal or better in terms of both operations.

Two Multi-Armed-Bandit Approaches As described in Sec. 3.4, our multi-armed bandit approach with dynamic mixing ratio during multi-task training learns a sufficiently good curriculum to improve the sentence simplification task (see Sec. 5). Here, we further show an ablation study on another alternative approach of using multi-armed bandits, where we record the last 10% of the actions from the bandit controller[15], then calculate the corresponding mixing ra-

[14]For this entailment analysis, we use ground-truth output as premise instead of input source, because: (1) entailment w.r.t. input source can give artificially high scores even when the output doesn't simplify enough and just copies the source (see the discussion in Sec. 5 and Table 4); (2) By transitivity, if output is entailed by ground-truth, which in turn is entailed by source, then output should also be entailed by source (plus, we want the output to be closer to ground-truth than to input source).

[15]We choose the last 10% to avoid the noisy action-value estimates at the start of the training.

tio based on this 10%, and run another independent model from scratch with this fixed mixing ratio. We found that the curriculum-style dynamic switching of tasks is in fact very effective as compared to this other 2-stage approach (33.22 versus 32.58 in SARI with $p < 0.01$). This is intuitive because the dynamic switching of tasks during multi-task training allows the model to choose the best next task to run based on the current state (as well as the previous curriculum path) of the model, as opposed to a fixed/static single mixing ratio for the full training period. In Fig. 3, we visualize the (moving averages of) probabilities of selecting each task, which shows that in the 0-1000 #rounds range, the bandit initially gives higher weight to the main task, but gradually redistributes the probabilities to the auxiliary tasks; and beyond 1000 #rounds, it then alternates switching among the three different tasks periodically. We also experimented with replacing the bandit controller with random task choices, and our bandit-controller achieves statistically significantly better results than this approach in both SARI and FKGL with $p < 0.01$, which shows that the path learned by the bandit controller is meaningful.

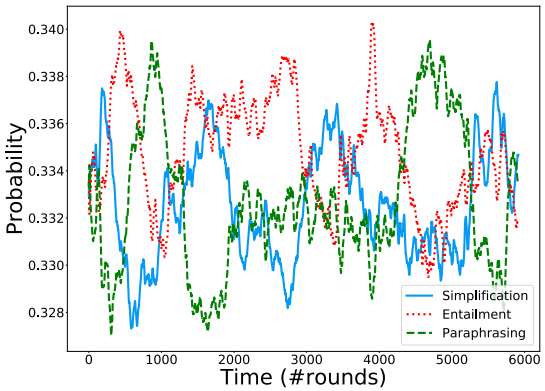

Figure 3: Task selection probability over training trajectory, predicted by bandit controller.

Multi-Task Learning vs. Data Augmentation To verify that our improvements come indeed from the auxiliary tasks' specific character/capabilities and not just due to adding more data, we separately trained word embeddings on each auxiliary dataset (i.e., SNLI+MultiNLI and ParaNMT) and incorporated them into the primary simplification model. We found that both our 2-way multi-task models perform stat. significantly better than these models (which use the auxiliary word-embeddings), suggesting that merely adding more data is not enough. Moreover, Table 5 shows that only specific intuitive (syntactic vs. semantic) layer sharing between the primary and auxiliary tasks helps results and not just adding data.

Output Examples Fig. 4 shows two output examples comparing DRESS-LS, pointer baseline, and multi-task models (and reference). We see that our multi-task model simplifies the input appropriately (similar extent to reference) while also keeping reasonably important information from the source. The pointer baseline and the DRESS-LS models simplify to a lesser extent and keep much more of the original input (as also suggested by our match-with-input investigation in Table 4).

Input: *he put henson in charge of escorting his slaves to his brother 's kentucky plantation .*
Reference: *he sent henson to take his slaves to kentucky .*
DRESS-LS: *he put henson in charge of escorting his slaves to his brother 's kentucky plantation .*
Baseline: *he put his slaves to his brother 's kentucky plantation .*
Multi-Task: *he put henson in charge of escorting .*

Input: *northern states did not allow slavery , but escaped slaves were returned to their owners as property , so henson would have to flee to canada to be free .*
Reference: *states in the north did not allow slavery .*
DRESS-LS: *southern states did not allow slavery , but the guatemalans were returned to their owners as property .*
Baseline: *he slaves were returned to their owners as property .*
Multi-Task: *northern states did not allow slavery .*

Figure 4: Output examples comparing DRESS-LS, our pointer baseline, and multi-task model.

7 Conclusion

We presented a multi-level, multi-task learning approach to incorporate natural language inference and paraphrasing knowledge into sentence simplification models, via soft sharing at higher-level semantic and lower-level lexico-syntactic levels. We also introduced a multi-armed bandits approach for learning a dynamic mixing ratio of tasks. We demonstrated strong simplification improvements on three standard datasets via automatic and human evaluation, and also discussed several ablation and analysis studies.

Acknowledgments

We thank the reviewers for their helpful comments (and Xingxing Zhang for providing preprocessed datasets). This work was supported by DARPA (YFA17-D17AP00022), Google Faculty Research Award, Bloomberg Data Science Research Grant, and NVidia GPU awards. The views contained in this article are those of the authors and not of the funding agency.

References

Marcelo Amancio and Lucia Specia. 2014. An analysis of crowdsourced text simplifications. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 123–130.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002a. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002b. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Regina Barzilay and Kathleen R McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th annual meeting on Association for Computational Linguistics*, pages 50–57. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *ACL (1)*, pages 1415–1425.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122.

John A Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *EACL*, pages 269–270.

Rich Caruana. 1998. Multitask learning. In *Learning to learn*, pages 95–133. Springer.

Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 1041–1044. Association for Computational Linguistics.

Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1657–1668.

James Clarke and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 377–384. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

William Coster and David Kauchak. 2011. Learning to simplify sentences using wikipedia. In *Proceedings of the workshop on monolingual text-to-text generation*, pages 1–9. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

Siobhan Lucy Devlin. 1999. *Simplifying natural language for aphasic readers.* Ph.D. thesis, University of Sunderland.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Richard Evans, Constantin Orasan, and Iustin Dornescu. 2014. An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 131–140.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1608–1618.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 25–32. Association for Computational Linguistics.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *HLT-NAACL*, pages 758–764.

Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 63–68.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. *arXiv preprint arXiv:1704.03003*.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft, layer-specific multi-task summarization with entailment and question generation. In *Proceedings of ACL*.

Sanda Harabagiu and Andrew Hickl. 2006. Methods for using textual entailment in open-domain question answering. In *ACL*, pages 905–912.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *EMNLP*.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. In *ACL (2)*, pages 458–463.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. 2016. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from standard wikipedia to simple wikipedia. In *NAACL-HLT*, pages 211–217.

Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: a project note. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 9–16. Association for Computational Linguistics.

Sergio Jimenez, George Duenas, Julia Baquero, Alexander Gelbukh, Av Juan Dios Bátiz, and Av Mendizábal. 2014. UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *In SemEval*, pages 732–742.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017. One model to learn them all. *arXiv preprint arXiv:1706.05137*.

Nobuhiro Kaji, Daisuke Kawahara, Sadao Kurohash, and Satoshi Sato. 2002. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 215–222. Association for Computational Linguistics.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *ACL (1)*, pages 1537–1546.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

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

Beata Beigman Klebanov, Kevin Knight, and Daniel Marcu. 2004. Text simplification for information-seeking applications. *Lecture Notes in Computer Science*, pages 735–747.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Alice Lai and Julia Hockenmaier. 2014. Illinois-lh: A denotational and distributional approach to semantics. *Proc. SemEval*, 2:5.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *ACL*.

Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Ramakanth Pasunuru and Mohit Bansal. 2017. Multi-task video captioning with video and entailment generation. In *Proceedings of ACL*.

Ramakanth Pasunuru, Han Guo, and Mohit Bansal. 2017. Towards improving abstractive summarization via entailment generation. In *NFiS@EMNLP*.

Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *Workshop on Speech and Language Technology in Education*.

Luz Rello, Ricardo Baeza-Yates, and Horacio Saggion. 2013. The impact of lexical simplification by verbal paraphrases for people with and without dyslexia. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 501–512. Springer.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2017. Sluice networks: Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Sahil Sharma and Balaraman Ravindran. 2017. Online multi-task learning using active sampling. *CoRR*, abs/1702.06053.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.

Advaith Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.

Lucia Specia. 2010. Translating from complex to simplified sentences. *Computational Processing of the Portuguese Language*, pages 30–39.

Sanja Štajner, Hannah Béchara, and Horacio Saggion. 2015. A deeper exploration of the standard pb-smt approach to text simplification and its evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *ACL*, pages 344–352.

John Wieting and Kevin Gimpel. 2017a. Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *CoRR*, abs/1711.05732.

John Wieting and Kevin Gimpel. 2017b. Revisiting recurrent networks for paraphrastic sentence embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the conference on empirical methods in natural language processing*, pages 409–420. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2014. Text rewriting improves semantic role labeling. *Journal of Artificial Intelligence Research*, 51:133–164.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *ACL*.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association of Computational Linguistics*, 3(1):283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.

Congle Zhang, Stephen Soderland, and Daniel S Weld. 2015. Exploiting parallel news streams for unsupervised event extraction. *Transactions of the Association for Computational Linguistics*, 3:117–129.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1353–1361. Association for Computational Linguistics.

A Appendix

A.1 Training Details

All LSTMs use hidden state size of 256. We train word vectors with embedding size of 128 with random initialization. We use gradient clipped norm of 2.0. Our model selection (tuning) criteria is based on the average of our 3 metrics (SARI, BLEU, 1/FKGL) on the validation set. The mixing ratios are α_{ss}:α_{eg}:α_{pp} = 6:1:3 for Newsela, 6:1:3 for WikiSmall, and 7:2:1 for WikiLarge. The soft-sharing coefficient λ is set such that we balance the cross-entropy and regularization losses (at convergence), which is 5×10^{-6} for Newsela, 1×10^{-6} WikiSmall, and 1×10^{-5} for WikiLarge. We train models from scratch for Newsela and WikiSmall (using Adam (Kingma and Ba, 2014) optimizer with learning rate of 0.002 and 0.0015, respectively). However, because of the large size and computation overhead for WikiLarge, we first pre-train both main and auxiliary models on their own domain until they reach 90%

convergence, and use these models to initialize the multi-task models, and set the learning rate to $1/10$ of its original default value (0.001). We set the decay rate α in the bandit controller to be 0.3. We use the negative validation loss as the reward at each sampling step to the bandit algorithm. The validation loss is divided by two as a smoothing technique.[16] All our soft/hard and layer-specific sharing decisions (Sec. 6) were made on the validation/dev set. We follow previous work (Zhang and Lapata, 2017) in their pre-processing and post-processing of named entities. We capped vocabulary size to be 50K and replaced less frequent words with UNK token.[17] Unlike previous work (Zhang and Lapata, 2017), we do not use UNK-replacement at test time, but instead rely on our pointer-copy mechanism. We use beam search with beam size of 5. All other details provided in our released code.

[16] This constant serves the same purpose as the temperature variable in the softmax function.

[17] We measured the vocabulary overlap between the main and auxiliary tasks, and found that "word-form-overlap" (percentage of unique word types in auxiliary task that also appear in the main task) to be 40.7% (entailment) and 41.0% (paraphrase), and "word-count-overlap" (percentage of words in auxiliary task that also appear in the main task, based on token frequency counts) to be 95.2% (entailment) and 94.9% (paraphrase). Hence, this suggests that only rare words (which make up for very few counts) aren't considered in training process, and our pointer mechanism handles these extra UNK words by copying the actual word-form from the source to the output.