Soft Layer-Specific Multi-Task Summarization with Entailment and Question Generation

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

Abstract

An accurate abstractive summary of a document should contain all its salient information and should be logically entailed by the input document. We improve these important aspects of abstractive summarization via multi-task learning with the auxiliary tasks of question generation and entailment generation, where the former teaches the summarization model how to look for salient questioning-worthy details, and the latter teaches the model how to rewrite a summary which is a directed-logical subset of the input document. We also propose novel multi-task architectures with high-level (semantic) layer-specific sharing across multiple encoder and decoder layers of the three tasks, as well as soft-sharing mechanisms (and show performance ablations and analysis examples of each contribution). Overall, we achieve statistically significant improvements over the state-of-the-art on both the CNN/DailyMail and Gigaword datasets, as well as on the DUC-2002 transfer setup. We also present several quantitative and qualitative analysis studies of our model's learned saliency and entailment skills.

1 Introduction

Abstractive summarization is the challenging NLG task of compressing and rewriting a document into a short, relevant, salient, and coherent summary. It has numerous applications such as summarizing storylines, event understanding, etc. As compared to extractive or compressive summarization (Jing and McKeown, 2000; Knight and Marcu, 2002; Clarke and Lapata, 2008; Filippova et al., 2015; Henß et al., 2015), abstractive summaries are based on rewriting as opposed to selecting. Recent end-to-end, neural sequence-to-sequence models and larger datasets have allowed substantial progress on the abstractive task, with ideas ranging from copy-pointer mechanism and redundancy coverage, to metric reward based reinforcement learning (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; See et al., 2017).

Despite these strong recent advancements, there is still a lot of scope for improving the summary quality generated by these models. A good rewritten summary is one that contains all the salient information from the document, is logically followed (entailed) by it, and avoids redundant information. The redundancy aspect was addressed by coverage models (Suzuki and Nagata, 2016; Chen et al., 2016; Nallapati et al., 2016; See et al., 2017), but we still need to teach these models about how to better detect salient information from the input document, as well as about better logically-directed natural language inference skills.

In this work, we improve abstractive text summarization via soft, high-level (semantic) layer-specific multi-task learning with two relevant auxiliary tasks. The first is that of document-to-question generation, which teaches the summarization model about what are the right questions to ask, which in turn is directly related to what the salient information in the input document is. The second auxiliary task is a premise-to-entailment generation task to teach it how to rewrite a summary which is a directed-logical subset of (i.e., logically follows from) the input document, and contains no contradictory or unrelated information. For the question generation task, we use the SQuAD dataset (Rajpurkar et al., 2016), where we learn to generate a question given a sentence containing the answer, similar to the recent work

* Equal contribution.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 687–697
Melbourne, Australia, July 15 - 20, 2018.

by Du et al. (2017). Our entailment generation task is based on the recent SNLI classification dataset and task (Bowman et al., 2015), converted to a generation task (Pasunuru and Bansal, 2017).

Further, we also present novel multi-task learning architectures based on multi-layered encoder and decoder models, where we empirically show that it is substantially better to share the higher-level semantic layers between the three aforementioned tasks, while keeping the lower-level (lexico-syntactic) layers unshared. We also explore different ways to optimize the shared parameters and show that 'soft' parameter sharing achieves higher performance than hard sharing.

Empirically, our soft, layer-specific sharing model with the question and entailment generation auxiliary tasks achieves statistically significant improvements over the state-of-the-art on both the CNN/DailyMail and Gigaword datasets. It also performs significantly better on the DUC-2002 transfer setup, demonstrating its strong generalizability as well as the importance of auxiliary knowledge in low-resource scenarios. We also report improvements on our auxiliary question and entailment generation tasks over their respective previous state-of-the-art. Moreover, we significantly decrease the training time of the multi-task models by initializing the individual tasks from their pretrained baseline models. Finally, we present human evaluation studies as well as detailed quantitative and qualitative analysis studies of the improved saliency detection and logical inference skills learned by our multi-task model.

2 Related Work

Automatic text summarization has been progressively improving over the time, initially more focused on extractive and compressive models (Jing and McKeown, 2000; Knight and Marcu, 2002; Clarke and Lapata, 2008; Filippova et al., 2015; Kedzie et al., 2015), and moving more towards compressive and abstractive summarization based on graphs and concept maps (Giannakopoulos, 2009; Ganesan et al., 2010; Falke and Gurevych, 2017) and discourse trees (Gerani et al., 2014), syntactic parse trees (Cheung and Penn, 2014; Wang et al., 2013), and Abstract Meaning Representations (AMR) (Liu et al., 2015; Dohare and Karnick, 2017). Recent work has also adopted machine translation inspired neural seq2seq models for abstractive summarization with advances in hierarchical, distractive, saliency, and graph-attention modeling (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; Chen et al., 2016; Tan et al., 2017). Paulus et al. (2018) and Henß et al. (2015) incorporated recent advances from reinforcement learning. Also, See et al. (2017) further improved results via pointer-copy mechanism and addressed the redundancy with coverage mechanism.

Multi-task learning (MTL) is a useful paradigm to improve the generalization performance of a task with related tasks while sharing some common parameters/representations (Caruana, 1998; Argyriou et al., 2007; Kumar and Daumé III, 2012). Several recent works have adopted MTL in neural models (Luong et al., 2016; Misra et al., 2016; Hashimoto et al., 2017; Pasunuru and Bansal, 2017; Ruder et al., 2017; Kaiser et al., 2017). Moreover, some of the above works have investigated the use of shared vs unshared sets of parameters. On the other hand, we investigate the importance of soft parameter sharing and high-level versus low-level layer-specific sharing.

Our previous workshop paper (Pasunuru et al., 2017) presented some preliminary results for multi-task learning of textual summarization with entailment generation. This current paper has several major differences: (1) We present question generation as an additional effective auxiliary task to enhance the important complementary aspect of saliency detection; (2) Our new high-level layer-specific sharing approach is significantly better than alternative layer-sharing approaches (including the decoder-only sharing by Pasunuru et al. (2017)); (3) Our new soft sharing parameter approach gives stat. significant improvements over hard sharing; (4) We propose a useful idea of starting multi-task models from their pretrained baselines, which significantly speeds up our experiment cycle[1]; (5) For evaluation, we show diverse improvements of our soft, layer-specific MTL model (over state-of-the-art pointer+coverage baselines) on the CNN/DailyMail, Gigaword, as well as DUC datasets; we also report human evaluation plus analysis examples of learned saliency and entailment skills; we also report improvements on the auxiliary question and entailment generation tasks over their respective previous state-of-the-art.

[1] About 4-5 days for Pasunuru et al. (2017) approach vs. only 10 hours for us. This will allow the community to try many more multi-task training and tuning ideas faster.

In our work, we use a question generation task to improve the saliency of abstractive summarization in a multi-task setting. Using the SQuAD dataset (Rajpurkar et al., 2016), we learn to generate a question given the sentence containing the answer span in the comprehension (similar to Du et al. (2017)). For the second auxiliary task of entailment generation, we use the generation version of the RTE classification task (Dagan et al., 2006; Lai and Hockenmaier, 2014; Jimenez et al., 2014; Bowman et al., 2015). Some previous work has explored the use of RTE for redundancy detection in summarization by modeling graph-based relationships between sentences to select the most non-redundant sentences (Mehdad et al., 2013; Gupta et al., 2014), whereas our approach is based on multi-task learning.

3 Models

First, we introduce our pointer+coverage baseline model and then our two auxiliary tasks: question generation and entailment generation (and finally the multi-task learning models in Sec. 4).

3.1 Baseline Pointer+Coverage Model

We use a sequence-attention-sequence model with a 2-layer bidirectional LSTM-RNN encoder and a 2-layer uni-directional LSTM-RNN decoder, along with Bahdanau et al. (2015) style attention. Let $x = \{x_1, x_2, ..., x_m\}$ be the source document and $y = \{y_1, y_2, ..., y_n\}$ be the target summary. The output summary generation vocabulary distribution conditioned over the input source document is $P_v(y|x;\theta) = \prod_{t=1}^{n} p_v(y_t|y_{1:t-1}, x;\theta)$. Let the decoder hidden state be s_t at time step t and let c_t be the context vector which is defined as a weighted combination of encoder hidden states. We concatenate the decoder's (last) RNN layer hidden state s_t and context vector c_t and apply a linear transformation, and then project to the vocabulary space by another linear transformation. Finally, the conditional vocabulary distribution at each time step t of the decoder is defined as:

$$p_v(y_t|y_{1:t-1}, x;\theta) = \mathrm{sfm}(V_p(W_f[s_t; c_t]+b_f)+b_p) \quad (1)$$

where, W_f, V_p, b_f, b_p are trainable parameters, and $\mathrm{sfm}(\cdot)$ is the softmax function.

Pointer-Generator Networks Pointer mechanism (Vinyals et al., 2015) helps in directly copying the words from the source sequence during target sequence generation, which is a good fit for a task like summarization. Our pointer mechanism approach is similar to See et al. (2017), who use a soft switch based on the generation probability $p_g = \sigma(W_g c_t + U_g s_t + V_g e_{w_{t-1}} + b_g)$, where $\sigma(\cdot)$ is a sigmoid function, W_g, U_g, V_g and b_g are parameters learned during training. $e_{w_{t-1}}$ is the previous time step output word embedding. The final word distribution is $P_f(y) = p_g \cdot P_v(y) + (1-p_g) \cdot P_c(y)$, where P_v vocabulary distribution is as shown in Eq. 1, and copy distribution P_c is based on the attention distribution over source document words.

Coverage Mechanism Following previous work (See et al., 2017), coverage helps alleviate the issue of word repetition while generating long summaries. We maintain a coverage vector $\hat{c}_t = \sum_{t=0}^{t-1} \alpha_t$ that sums over all of the previous time steps attention distributions α_t, and this is added as input to the attention mechanism. Coverage loss is $L_{cov}(\theta) = \sum_t \sum_i min(\alpha_{t,i}, \hat{c}_{t,i})$. Finally, the total loss is a weighted combination of cross-entropy loss and coverage loss:

$$L(\theta) = -\log P_f(y) + \lambda L_{cov}(\theta) \quad (2)$$

where λ is a tunable hyperparameter.

3.2 Two Auxiliary Tasks

Despite the strengths of the strong model described above with attention, pointer, and coverage, a good summary should also contain maximal salient information and be a directed logical entailment of the source document. We teach these skills to the abstractive summarization model via multi-task training with two related auxiliary tasks: question generation task and entailment generation.

Question Generation The task of question generation is to generate a question from a given input sentence, which in turn is related to the skill of being able to find the important salient information to ask questions about. First the model has to identify the important information present in the given sentence, then it has to frame (generate) a question based on this salient information, such that, given the sentence and the question, one has to be able to predict the correct answer (salient information in this case). A good summary should also be able to find and extract all the salient information in the given source document, and hence we incorporate such capabilities into our abstractive text summarization model by multi-task

learning it with a question generation task, sharing some common parameters/representations (see more details in Sec. 4). For setting up the question generation task, we follow Du et al. (2017) and use the SQuAD dataset to extract sentence-question pairs. Next, we use the same sequence-to-sequence model architecture as our summarization model. Note that even though our question generation task is generating one question at a time[2], our multi-task framework (see Sec. 4) is set up in such a way that the sentence-level knowledge from this auxiliary task can help the document-level primary (summarization) task to generate multiple salient facts – by sharing high-level semantic layer representations. See Sec. 7 and Table 10 for a quantitative evaluation showing that the multi-task model can find multiple (and more) salient phrases in the source document. Also see Sec. 7 (and supp) for challenging qualitative examples where baseline and SotA models only recover a small subset of salient information but our multi-task model with question generation is able to detect more of the important information.

Entailment Generation The task of entailment generation is to generate a hypothesis which is entailed by (or logically follows from) the given premise as input. In summarization, the generation decoder also needs to generate a summary that is entailed by the source document, i.e., does not contain any contradictory or unrelated/extraneous information as compared to the input document. We again incorporate such inference capabilities into the summarization model via multi-task learning, sharing some common representations/parameters between our summarization and entailment generation model (more details in Sec. 4). For this task, we use the entailment-labeled pairs from the SNLI dataset (Bowman et al., 2015) and set it up as a generation task (using the same strong model architecture as our abstractive summarization model). See Sec. 7 and Table 9 for a quantitative evaluation showing that the multi-task model is better entailed by the source document and has fewer extraneous facts. Also see Sec. 7 and supplementary for qualitative examples of how our multi-task model with the entailment auxiliary task is able to generate more logically-entailed summaries than the baseline and SotA models, which instead produce extraneous, unrelated words not present (in any paraphrased form) in the source document.

[2]We also tried to generate all the questions at once from the full document, but we obtained low accuracy because of this task's challenging nature and overall less training data.

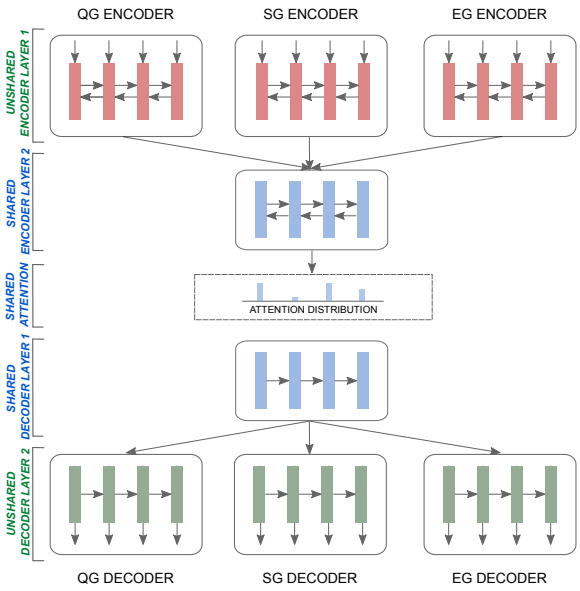

Figure 1: Overview of our multi-task model with parallel training of three tasks: abstractive summary generation (SG), question generation (QG), and entailment generation (EG). We share the 'blue' color representations across all the three tasks, i.e., second layer of encoder, attention parameters, and first layer of decoder.

4 Multi-Task Learning

We employ multi-task learning for parallel training of our three tasks: abstractive summarization, question generation, and entailment generation. In this section, we describe our novel layer-specific, soft-sharing approaches and other multi-task learning details.

4.1 Layer-Specific Sharing Mechanism

Simply sharing all parameters across the related tasks is not optimal, because models for different tasks have different input and output distributions, esp. for low-level vs. high-level parameters. Therefore, related tasks should share some common representations (e.g., high-level information), as well as need their own individual task-specific representations (esp. low-level information). To this end, we allow different components of model parameters of related tasks to be shared vs. unshared, as described next.

Encoder Layer Sharing: Belinkov et al. (2017) observed that lower layers (i.e., the layers closer to the input words) of RNN cells in a seq2seq

machine translation model learn to represent word structure, while higher layers (farther from input) are more focused on high-level semantic meanings (similar to findings in the computer vision community for image features (Zeiler and Fergus, 2014)). We believe that while textual summarization, question generation, and entailment generation have different training data distributions and low-level representations, they can still benefit from sharing their models' high-level components (e.g., those that capture the skills of saliency and inference). Thus, we keep the lower-level layer (i.e., first layer closer to input words) of the 2-layer encoder of all three tasks unshared, while we share the higher layer (second layer in our model as shown in Fig. 1) across the three tasks.

Decoder Layer Sharing: Similarly for the decoder, lower layers (i.e., the layers closer to the output words) learn to represent word structure for generation, while higher layers (farther from output) are more focused on high-level semantic meaning. Hence, we again share the higher level components (first layer in the decoder far from output as show in Fig. 1), while keeping the lower layer (i.e., second layer) of decoders of all three tasks unshared.

Attention Sharing: As described in Sec. 3.1, the attention mechanism defines an attention distribution over high-level layer encoder hidden states and since we share the second, high-level (semantic) layer of all the encoders, it is intuitive to share the attention parameters as well.

4.2 Soft vs. Hard Parameter Sharing

Hard-sharing: In the most common multi-task learning hard-sharing approach, the parameters to be shared are forced to be the same. As a result, gradient information from multiple tasks will directly pass through shared parameters, hence forcing a common space representation for all the related tasks. **Soft-sharing**: In our soft-sharing approach, we encourage shared parameters to be close in representation space by penalizing their l_2 distances. Unlike hard sharing, this approach gives more flexibility for the tasks by only loosely coupling the shared space representations. We minimize the following loss function for the primary task in soft-sharing approach:

$$L(\theta) = -\log P_f(y) + \lambda L_{cov}(\theta) + \gamma \|\theta_s - \psi_s\| \quad (3)$$

where γ is a hyperparameter, θ represents the primary summarization task's full parameters, while θ_s and ψ_s represent the shared parameter subset between the primary and auxiliary tasks.

4.3 Fast Multi-Task Training

During multi-task learning, we alternate the mini-batch optimization of the three tasks, based on a tunable 'mixing ratio' $\alpha_s : \alpha_q : \alpha_e$; i.e., optimizing the summarization task for α_s mini-batches followed by optimizing the question generation task for α_q mini-batches, followed by entailment generation task for α_e mini-batches (and for 2-way versions of this, we only add one auxiliary task at a time). We continue this process until all the models converge. Also, importantly, instead of training from scratch, we start the primary task (summarization) from a 90%-converged model of its baseline to make the training process faster. We observe that starting from a fully-converged baseline makes the model stuck in a local minimum. In addition, we also start all auxiliary models from their 90%-converged baselines, as we found that starting the auxiliary models from scratch has a chance to pull the primary model's shared parameters towards randomly-initialized auxiliary model's shared parameters.

5 Experimental Setup

Datasets: We use CNN/DailyMail dataset (Hermann et al., 2015; Nallapati et al., 2016) and Gigaword (Rush et al., 2015) datasets for summarization, and the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) and the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) datasets for our entailment and question generation tasks, resp. We also show generalizability/transfer results on DUC-2002 with our CNN/DM trained models. Supplementary contains dataset details.

Evaluation Metrics: We use the standard ROUGE evaluation package (Lin, 2004) for reporting the results on all of our summarization models. Following previous work (Chopra et al., 2016; Nallapati et al., 2016), we use ROUGE full-length F1 variant for all our results. Following See et al. (2017), we also report METEOR (Denkowski and Lavie, 2014) using the MS-COCO evaluation script (Chen et al., 2015).

Human Evaluation Criteria: We used Amazon MTurk to perform human evaluation of summary *relevance* and *readability*. We selected human annotators that were located in the US, had an ap-

Models	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
PREVIOUS WORK				
Seq2Seq(50k vocab) (See et al., 2017)	31.33	11.81	28.83	12.03
Pointer (See et al., 2017)	36.44	15.66	33.42	15.35
Pointer+Coverage (See et al., 2017) ⋆	39.53	17.28	36.38	18.72
Pointer+Coverage (See et al., 2017) †	38.82	16.81	35.71	18.14
OUR MODELS				
Two-Layer Baseline (Pointer+Coverage) ⊗	39.56	17.52	36.36	18.17
⊗ + Entailment Generation	39.84	17.63	36.54	18.61
⊗ + Question Generation	39.73	17.59	36.48	18.33
⊗ + Entailment Gen. + Question Gen.	39.81	17.64	36.54	18.54

Table 1: CNN/DailyMail summarization results. ROUGE scores are full length F-1 (as previous work). All the multi-task improvements are statistically significant over the state-of-the-art baseline.

Models	R-1	R-2	R-L
PREVIOUS WORK			
ABS+ (Rush et al., 2015)	29.76	11.88	26.96
RAS-El (Chopra et al., 2016)	33.78	15.97	31.15
lvt2k (Nallapati et al., 2016)	32.67	15.59	30.64
Pasunuru et al. (2017)	32.75	15.35	30.82
OUR MODELS			
2-Layer Pointer Baseline ⊗	34.26	16.40	32.03
⊗ + Entailment Generation	35.45	17.16	33.19
⊗ + Question Generation	35.48	17.31	32.97
⊗ + Entailment + Question	35.98	17.76	33.63

Table 2: Summarization results on Gigaword. ROUGE scores are full length F-1.

proval rate greater than 95%, and had at least 10,000 approved HITs. For the pairwise model comparisons discussed in Sec. 6.2, we showed the annotators the input article, the ground truth summary, and the two model summaries (randomly shuffled to anonymize model identities) – we then asked them to choose the better among the two model summaries or choose 'Not-Distinguishable' if both summaries are equally good/bad. Instructions for relevance were defined based on the summary containing salient/important information from the given article, being correct (i.e., avoiding contradictory/unrelated information), and avoiding redundancy. Instructions for readability were based on the summary's fluency, grammaticality, and coherence.

Training Details All our soft/hard and layer-specific sharing decisions were made on the validation/development set. Details of RNN hidden state sizes, Adam optimizer, mixing ratios, etc. are provided in the supplementary for reproducibility.

6 Results

6.1 Summarization (Primary Task) Results

Pointer+Coverage Baseline We start from the strong model of See et al. (2017).[3] Table 1 shows that our baseline model performs better than or comparable to See et al. (2017).[4] On Gigaword dataset, our baseline model (with pointer only, since coverage not needed for this single-sentence summarization task) performs better than all previous works, as shown in Table 2.

Multi-Task with Entailment Generation We first perform multi-task learning between abstractive summarization and entailment generation with soft-sharing of parameters as discussed in Sec. 4. Table 1 and Table 2 shows that this multi-task setting is better than our strong baseline models and the improvements are statistically significant on all metrics[5] on both CNN/DailyMail ($p < 0.01$ in ROUGE-1/ROUGE-L/METEOR and $p < 0.05$ in ROUGE-2) and Gigaword ($p < 0.01$ on all metrics) datasets, showing that entailment generation task is inducing useful inference skills to the summarization task (also see analysis examples in Sec. 7).

Multi-Task with Question Generation For multi-task learning with question generation, the improvements are statistically significant in ROUGE-1 ($p < 0.01$), ROUGE-L ($p < 0.05$), and METEOR ($p < 0.01$) for CNN/DailyMail and in all metrics ($p < 0.01$) for Gigaword, compared to the respective baseline models. Also, Sec. 7 presents quantitative and qualitative analysis of this model's improved saliency.[6]

[3] We use two layers so as to allow our high- versus low-level layer sharing intuition. Note that this does not increase the parameter size much (23M versus 22M for See et al. (2017)).

[4] As mentioned in the github for See et al. (2017), their publicly released pretrained model produces the lower scores that we represent by † in Table 1.

[5] Stat. significance is computed via bootstrap test (Noreen, 1989; Efron and Tibshirani, 1994) with 100K samples.

[6] In order to verify that our improvements were 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 and SQuAD) and incorporated them into the summarization model. We found that both our 2-way multi-task models perform sig-

Models	Relevance	Readability	Total
MTL VS. BASELINE			
MTL wins	43	40	83
Baseline wins	22	24	46
Non-distinguish.	35	36	71
MTL VS. SEE ET AL. (2017)			
MTL wins	39	33	72
See (2017) wins	29	38	67
Non-distinguish.	32	29	61

Table 3: CNN/DM Human Evaluation: pairwise comparison between our 3-way multi-task (MTL) model w.r.t. our baseline and See et al. (2017).

Models	Relevance	Readability	Total
MTL wins	33	32	65
Baseline wins	22	22	44
Non-distinguish.	45	46	91

Table 4: Gigaword Human Evaluation: pairwise comparison between our 3-way multi-task (MTL) model w.r.t. our baseline.

Multi-Task with Entailment and Question Generation Finally, we perform multi-task learning with all three tasks together, achieving the best of both worlds (inference skills and saliency). Table 1 and Table 2 show that our full multi-task model achieves the best scores on CNN/DailyMail and Gigaword datasets, and the improvements are statistically significant on all metrics on both CNN/DailyMail ($p < 0.01$ in ROUGE-1/ROUGE-L/METEOR and $p < 0.02$ in ROUGE-2) and Gigaword ($p < 0.01$ on all metrics). Finally, our 3-way multi-task model (with both entailment and question generation) outperforms the publicly-available pretrained result (†) of the previous SotA (See et al., 2017) with stat. significance ($p < 0.01$), as well the higher-reported results ($\star$) on ROUGE-1/ROUGE-2 ($p < 0.01$).

6.2 Human Evaluation

We also conducted a blind human evaluation on Amazon MTurk for relevance and readability, based on 100 samples, for both CNN/DailyMail and Gigaword (see instructions in Sec. 5). Table. 3 shows the CNN/DM results where we do pairwise comparison between our 3-way multi-task model's output summaries w.r.t. our baseline summaries and w.r.t. See et al. (2017) summaries. As shown, our 3-way multi-task model achieves both higher relevance and higher readability scores w.r.t. the baseline. W.r.t. See et al. (2017), our MTL model is higher in relevance scores but a bit lower in readability scores (and is higher in terms of total aggregate scores). One potential reason for this lower readability score is that our entailment generation auxiliary task encourages our summarization model to rewrite more and to be more abstractive than See et al. (2017) – see abstractiveness results in Table 11.

We also show human evaluation results on the Gigaword dataset in Table 4 (again based on pairwise comparisons for 100 samples), where we see that our MTL model is better than our state-of-the-art baseline on both relevance and readability.[7]

nificantly better than these models using the auxiliary word-embeddings, suggesting that merely adding more data in not enough.

Models	R-1	R-2	R-L
See et al. (2017)	34.30	14.25	30.82
Baseline	35.96	15.91	32.92
Multi-Task (EG + QG)	36.73	16.15	33.58

Table 5: ROUGE F1 scores on DUC-2002.

6.3 Generalizability Results (DUC-2002)

Next, we also tested our model's generalizability/transfer skills, where we take the models trained on CNN/DailyMail and directly test them on DUC-2002. We take our baseline and 3-way multi-task models, plus the pointer-coverage model from See et al. (2017).[8] We only retune the beam-size for each of these three models separately (based on DUC-2003 as the validation set).[9] As shown in Table 5, our multi-task model achieves statistically significant improvements over the strong baseline ($p < 0.01$ in ROUGE-1 and ROUGE-L) and the pointer-coverage model from See et al. (2017) ($p < 0.01$ in all metrics). This demonstrates that our model is able to generalize well and that the auxiliary knowledge helps more in low-resource scenarios.

6.4 Auxiliary Task Results

In this section, we discuss the individual/separated performance of our auxiliary tasks.

Entailment Generation We use the same architecture as described in Sec. 3.1 with pointer mech-

[7]Note that we did not have output files of any previous work's model on Gigaword; however, our baseline is already a strong state-of-the-art model as shown in Table 2.

[8]We use the publicly-available pretrained model from See et al. (2017)'s github for these DUC transfer results, which produces the † results in Table 1. All other comparisons and analysis in our paper are based on their higher $\star$ results.

[9]We follow previous work which has shown that larger beam values are better and feasible for DUC corpora. However, our MTL model still achieves stat. significant improvements ($p < 0.01$ in all metrics) over See et al. (2017) without beam retuning (i.e., with beam = 4).

Models	M	C	R	B
Pasunuru&Bansal (2017)	29.6	117.8	62.4	40.6
Our 1-layer pointer EG	32.4	139.3	65.1	43.6
Our 2-layer pointer EG	32.3	140.0	64.4	43.7

Table 6: Performance of our pointer-based entailment generation (EG) models compared with previous SotA work. M, C, R, B are short for Meteor, CIDEr-D, ROUGE-L, and BLEU-4, resp.

Models	M	C	R	B
Du et al. (2017)	15.2	-	38.0	10.8
Our 1-layer pointer QG	15.4	75.3	36.2	9.2
Our 2-layer pointer QG	17.5	95.3	40.1	13.8

Table 7: Performance of our pointer-based question generation (QG) model w.r.t. previous work.

anism, and Table 6 compares our model's performance to Pasunuru and Bansal (2017). Our pointer mechanism gives a performance boost, since the entailment generation task involves copying from the given premise sentence, whereas the 2-layer model seems comparable to the 1-layer model. Also, the supplementary shows some output examples from our entailment generation model.

Question Generation Again, we use same architecture as described in Sec. 3.1 along with pointer mechanism for the task of question generation. Table 7 compares the performance of our model w.r.t. the state-of-the-art Du et al. (2017). Also, the supplementary shows some output examples from our question generation model.

7 Ablation and Analysis Studies

Soft-sharing vs. Hard-sharing As described in Sec. 4.2, we choose soft-sharing over hard-sharing because of the more expressive parameter sharing it provides to the model. Empirical results in Table. 8 prove that soft-sharing method is statistically significantly better than hard-sharing with $p < 0.001$ in all metrics.[10]

Comparison of Different Layer-Sharing Methods We also conducted ablation studies among various layer-sharing approaches. Table 8 shows results for soft-sharing models with decoder-only sharing (D1+D2; similar to Pasunuru et al. (2017)) as well as lower-layer sharing (encoder layer 1 + decoder layer 2, with and without attention shared). As shown, our final model (high-level semantic layer sharing E2+Attn+D1) outperforms

[10]In the interest of space, most of the analyses are shown for CNN/DailyMail experiments, but we observed similar trends for the Gigaword experiments as well.

Models	R-1	R-2	R-L	M
Final Model	**39.81**	**17.64**	**36.54**	**18.54**
SOFT-VS.-HARD SHARING				
Hard-sharing	39.51	17.44	36.33	18.21
LAYER SHARING METHODS				
D1+D2	39.62	17.49	36.44	18.34
E1+D2	39.51	17.51	36.37	18.15
E1+Attn+D2	39.32	17.36	36.11	17.88

Table 8: Ablation studies comparing our final multi-task model with hard-sharing and different alternative layer-sharing methods. Here E1, E2, D1, D2, Attn refer to parameters of the first/second layer of encoder/decoder, and attention parameters. Improvements of final model upon ablation experiments are all stat. signif. with $p < 0.05$.

Models	Average Entailment Probability
Baseline	0.907
Multi-Task (EG)	0.912

Table 9: Entailment classification results of our baseline vs. EG-multi-task model ($p < 0.001$).

these alternate sharing methods in all metrics with statistical significance ($p < 0.05$).[11]

Quantitative Improvements in Entailment We employ a state-of-the-art entailment classifier (Chen et al., 2017), and calculate the average of the entailment probability of each of the output summary's sentences being entailed by the input source document. We do this for output summaries of our baseline and 2-way-EG multi-task model (with entailment generation). As can be seen in Table 9, our multi-task model improves upon the baseline in the aspect of being entailed by the source document (with statistical significance $p < 0.001$). Further, we use the Named Entity Recognition (NER) module from CoreNLP (Manning et al., 2014) to compute the number of times the output summary contains extraneous facts (i.e., named entities as detected by the NER system) that are not present in the source documents, based on the intuition that a well-entailed summary should not contain unrelated information not followed from the input premise. We found that our 2-way MTL model with entailment generation reduces this extraneous count by 17.2% w.r.t. the baseline. The qualitative examples below further discuss this issue of generating unrelated information.

Quantitative Improvements in Saliency Detection For our saliency evaluation, we used the

[11]Note that all our soft and layer sharing decisions were strictly made on the dev/validation set (see Sec. 5).

Models	Average Match Rate
Baseline	27.75 %
Multi-Task (QG)	28.06 %

Table 10: Saliency classification results of our baseline vs. QG-multi-task model ($p < 0.01$).

Models	2-gram	3-gram	4-gram
See et al. (2017)	2.24	6.03	9.72
MTL (3-way)	2.84	6.83	10.66

Table 11: Abstractiveness: novel n-gram percent.

answer-span prediction classifier from Pasunuru and Bansal (2018) trained on SQuAD (Rajpurkar et al., 2016) as the keyword detection classifier. We then annotate the ground-truth and model summaries with this keyword classifier and compute the % match, i.e., how many salient words from the ground-truth summary were also generated in the model summary. The results are shown in Table 10, where the 2-way-QG MTL model (with question generation) versus baseline improvement is stat. significant ($p < 0.01$). Moreover, we found 93 more cases where our 2-way-QG MTL model detects 2 or more additional salient keywords than the pointer baseline model (as opposed to vice versa), showing that sentence-level question generation task is helping the document-level summarization task in finding more salient terms.

Qualitative Examples on Entailment and Saliency Improvements Fig. 2 presents an example of output summaries generated by See et al. (2017), our baseline, and our 3-way multi-task model. See et al. (2017) and our baseline models generate phrases like "john hartson" and "hampden injustice" that don't appear in the input document, hence they are not entailed by the input.[12] Moreover, both models missed salient information like "josh meekings", "leigh griffiths", and "hoops", that our multi-task model recovers.[13] Hence, our 3-way multi-task model generates summaries that are both better at logical entailment and contain more salient information. We refer to supplementary Fig. 3 for more details and similar examples for separated 2-way multi-task models (supplementary Fig. 1, Fig. 2).

Abstractiveness Analysis As suggested in See et al. (2017), we also compute the abstractiveness score as the number of novel n-grams between the model output summary and source document. As shown in Table 11, our multi-task model (EG + QG) is more abstractive than See et al. (2017).

[12]These extra, non-entailed unrelated/contradictory information are not present at all in any paraphrase form in the input document.

[13]We consider the fill-in-the-blank highlights annotated by human on CNN/DailyMail dataset as salient information.

Figure 2: Example summary from our 3-way MTL model. The boxed-red highlights are extraneously-generated words not present/paraphrased in the input document. The unboxed-green highlights show salient phrases.

8 Conclusion

We presented a multi-task learning approach to improve abstractive summarization by incorporating the ability to detect salient information and to be logically entailed by the document, via question generation and entailment generation auxiliary tasks. We propose effective soft and high-level (semantic) layer-specific parameter sharing and achieve significant improvements over the state-of-the-art on two popular datasets, as well as a generalizability/transfer DUC-2002 setup.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported by DARPA (YFA17-D17AP00022), Google Faculty Research Award, Bloomberg Data Science Research Grant, and NVidia GPU awards. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the funding agency.

References

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2007. Multi-task feature learning. In *NIPS*.

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

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

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

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

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.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In *IJCAI*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Jackie Chi Kit Cheung and Gerald Penn. 2014. Unsupervised sentence enhancement for automatic summarization. In *EMNLP*, pages 775–786.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *HLT-NAACL*.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

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.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *EACL*.

Shibhansh Dohare and Harish Karnick. 2017. Text summarization using abstract meaning representation. *arXiv preprint arXiv:1706.01678*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *ACL*.

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

Tobias Falke and Iryna Gurevych. 2017. Bringing structure into summaries: Crowdsourcing a benchmark corpus of concept maps. In *EMNLP*.

Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *EMNLP*, pages 360–368.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*, pages 340–348. ACL.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *EMNLP*, volume 14, pages 1602–1613.

George Giannakopoulos. 2009. Automatic summarization from multiple documents. *Ph. D. dissertation*.

Anand Gupta, Manpreet Kaur, Adarsh Singh, Aseem Goel, and Shachar Mirkin. 2014. Text summarization through entailment-based minimum vertex cover. *Lexical and Computational Semantics (* SEM 2014)*, page 75.

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*.

Stefan Henß, Margot Mieskes, and Iryna Gurevych. 2015. A reinforcement learning approach for adaptive single-and multi-document summarization. In *GSCL*, pages 3–12.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

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 *SemEval*, pages 732–742.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and paste based text summarization. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 178–185, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Chris Kedzie, Kathleen McKeown, and Fernando Diaz. 2015. Predicting salient updates for disaster summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1608–1617.

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

Abhishek Kumar and Hal Daumé III. 2012. Learning task grouping and overlap in multi-task learning. In *ICML*.

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. In *Text Summarization Branches Out: Proceedings of the ACL-04 workshop*, volume 8.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward abstractive summarization using semantic representations. In *NAACL: HLT*, pages 1077–1086.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *ICLR*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Yashar Mehdad, Giuseppe Carenini, Frank W Tompa, and Raymond T Ng. 2013. Abstractive meeting summarization with entailment and fusion. In *Proc. of the 14th European Workshop on Natural Language Generation*, pages 136–146.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*.

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

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

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *NAACL*.

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

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *ICLR*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Sogaard. 2017. Sluice networks: Learning what to share between loosely related tasks. *CoRR*, abs/1705.08142.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*.

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

Jun Suzuki and Masaaki Nagata. 2016. Rnn-based encoder-decoder approach with word frequency estimation. In *EACL*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *ACL*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *NIPS*, pages 2692–2700.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *ACL*.

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