Towards Improving Abstractive Summarization via Entailment Generation

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

Abstract

Abstractive summarization, the task of rewriting and compressing a document into a short summary, has achieved considerable success with neural sequence-to-sequence models. However, these models can still benefit from stronger natural language inference skills, since a correct summary is logically entailed by the input document, i.e., it should not contain any contradictory or unrelated information. We incorporate such knowledge into an abstractive summarization model via multi-task learning, where we share its decoder parameters with those of an entailment generation model. We achieve promising initial improvements based on multiple metrics and datasets (including a test-only setting). The domain mismatch between the entailment (captions) and summarization (news) datasets suggests that the model is learning some domain-agnostic inference skills.

1 Introduction

Abstractive summarization, the task of rewriting a document into a short summary is a significantly more challenging (and natural) task than extractive summarization, which only involves choosing which sentence from the original document to keep or discard in the output summary. Neural sequence-to-sequence models have led to substantial improvements on this task of abstractive summarization, via machine translation inspired encoder-aligner-decoder approaches, further enhanced via convolutional encoders, pointer-copy mechanisms, and hierarchical attention (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017).

Input Document: *may is a pivotal month for moving and storage companies .*
Ground-truth Summary: *moving companies hit bumps in economic road*
Baseline Summary: *a month to move storage companies*
Multi-task Summary: *pivotal month for storage firms*

Figure 1: Motivating output example from our summarization+entailment multi-task model.

Despite these promising recent improvements, there is still scope in better teaching summarization models about the general natural language inference skill of logical entailment generation. This is because the task of abstractive summarization involves two subtasks: salient (important) event detection as well as logical compression, i.e., the summary should not contain any information that is contradictory or unrelated to the original document. Current methods have to learn both these skills from the same dataset and a single model. Therefore, there is benefit in learning the latter ability of logical compression via external knowledge from a separate entailment generation task, that will specifically teach the model how to rewrite and compress a sentence such that it logically follows from the original input.

To achieve this, we employ the recent paradigm of sequence-to-sequence multi-task learning (Luong et al., 2016). We share the decoder parameters of the summarization model with those of the entailment-generation model, so as to generate summaries that are good at both extracting important facts from as well as being logically entailed by the input document. Fig. 1 shows such an (actual) output example from our model, where it successfully learns both salient information extraction as well as entailment, unlike the strong baseline model.

Empirically, we report promising initial improvements over some solid baselines based on several metrics, and on multiple datasets: Gigaword and also a test-only setting of DUC. Importantly, these improvements are achieved despite the fact that the domain of the entailment dataset (image captions) is substantially different from the domain of the summarization datasets (general news), which suggests that the model is learning certain domain-independent inference skills. Our next steps to this workshop paper include incorporating stronger pointer-based models and employing the new multi-domain entailment corpus (Williams et al., 2017).

2 Related Work

Earlier summarization work focused more towards extractive (and compression) based summarization, i.e., selecting which sentences to keep vs discard, and also compressing based on choosing grammatically correct sub-sentences having the most important pieces of information (Jing, 2000; Knight and Marcu, 2002; Clarke and Lapata, 2008; Filippova et al., 2015). Bigger datasets and neural models have allowed the addressing of the complex reasoning involved in abstractive summarization, i.e., rewriting and compressing the input document into a new summary. Several advances have been made in this direction using machine translation inspired encoder-aligner-decoder models, convolution-based encoders, switching pointer and copy mechanisms, and hierarchical attention models (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017).

Recognizing textual entailment (RTE) is the classification task of predicting whether the relationship between a premise and hypothesis sentence is that of entailment (i.e., logically follows), contradiction, or independence (Dagan et al., 2006). The SNLI corpus Bowman et al. (2015) allows training accurate end-to-end neural networks for this task. Some previous work (Mehdad et al., 2013; Gupta et al., 2014) has explored the use of textual entailment recognition for redundancy detection in summarization. They label relationships between sentences, so as to select the most informative and non-redundant sentences for summarization, via sentence connectivity and graph-based optimization and fusion. Our focus, on the other hand, is entailment generation and not recognition, i.e., to teach summarization models the general natural language inference skill of generating a compressed sentence that logically entails the original longer sentence, so as to produce more effective short summaries. We achieve this via multi-task learning with entailment generation.

Multi-task learning involves sharing parameters between related tasks, whereby each task benefits from extra information in the training signals of the related tasks, and also improves its generalization performance. Luong et al. (2016) showed improvements on translation, captioning, and parsing in a shared multi-task setting. Recently, Pasunuru and Bansal (2017) extend this idea to video captioning with two related tasks: video completion and entailment generation. We demonstrate that abstractive text summarization models can also be improved by sharing parameters with an entailment generation task.

3 Models

First, we discuss our baseline model which is similar to the machine translation encoder-aligner-decoder model of Luong et al. (2015), and presented by Chopra et al. (2016). Next, we introduce our multi-task learning approach of sharing the parameters between abstractive summarization and entailment generation models.

3.1 Baseline Model

Our baseline model is a strong, multi-layered encoder-attention-decoder model with bilinear attention, similar to Luong et al. (2015) and following the details in Chopra et al. (2016). Here, we encode the source document with a two-layered LSTM-RNN and generate the summary using another two-layered LSTM-RNN decoder. The word probability distribution at time step t of the decoder is defined as follows:

$$p(w_t|w_{<t}, c_t, s_t) = softmax(W_s g(c_t, s_t)) \quad (1)$$

where g is a non-linear function and c_t and s_t are the context vector and LSTM-RNN decoder hidden state at time step t, respectively. The context vector $c_t = \sum \alpha_{t,i} h_i$ is a weighted combination of encoder hidden states h_i, where the attention weights are learned through the bilinear attention mechanism proposed in Luong et al. (2015). For the rest of the paper, we use same notations.

We also use the same model architecture for the entailment generation task, i.e., a sequence-to-sequence model encoding the premise and decoding the entailed hypothesis, via bilinear attention between them.

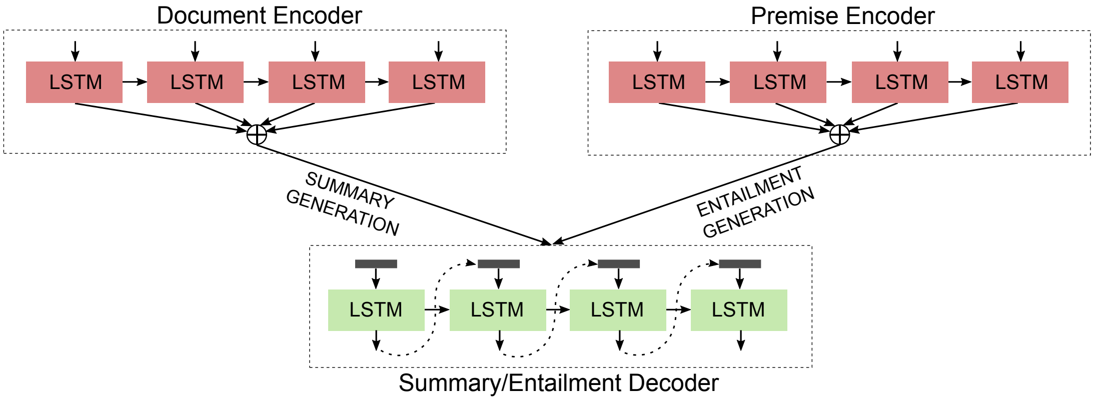

Figure 2: Multi-task learning of the summarization task (left) with the entailment generation task (right).

3.2 Multi-Task Learning

Multi-task learning helps in sharing knowledge between related tasks across domains (Luong et al., 2015). In this work, we show improvements on the task of abstractive summarization by sharing its parameters with the task of entailment generation. Since a summary is entailed by the input document, sharing parameters with the entailment generation task improves the logically-directed aspect of the summarization model, while maintaining the salient information extraction aspect.

In our multi-task setup, we share the decoder parameters of both the tasks (along with the word embeddings), as shown in Fig. 2, and we optimize the two loss functions (one for summarization and another for entailment generation) in alternate mini-batches of training. Let α_s be the number of mini-batches of training for summarization after which it is switched to train α_e number of mini-batches for entailment generation. Then, the mixing ratio is defined as $\frac{\alpha_s}{\alpha_s+\alpha_e} : \frac{\alpha_e}{\alpha_s+\alpha_e}$.

4 Experimental Setup

4.1 Datasets

Gigaword Corpus We use the exact annotated Gigaword corpus provided by Rush et al. (2015). The dataset has approximately 3.8 million training pairs. We use $10,000$ pairs as validation set and the exact test sample provided by Rush et al. (2015) as our test set. We use the first sentence of the article as the source with vocabulary size of $119,505$ and article headline as target with vocabulary size of $68,885$.

DUC Test Corpus The DUC corpus[1] comes in two variants: DUC-2003 corpus consists of 624 documents and DUC-2004 corpus consists of 500 documents. Each document in these datasets has four human annotated summaries. For experiments on this corpus, we directly used the Gigaword-trained model and tested on the DUC-2004 corpus. This is similar to the setups of Nallapati et al. (2016) and Chopra et al. (2016) (whereas the Rush et al. (2015) setup tunes on the DUC-2003 corpus).

SNLI corpus For the task of entailment generation, we use the Standford Natural Language Inference (SNLI) corpus (Bowman et al., 2015), where we only use the entailment-labeled pairs and regroup the splits to have a zero overlap train-test split and have a multi-reference test set, as suggested by Pasunuru and Bansal (2017). Out of $190,113$ entailments pairs, we use $145,822$ unique premise pairs for training, and the rest of them are equally divided into dev and test sets.

4.2 Evaluation

Following previous work (Nallapati et al., 2016; Chopra et al., 2016; Rush et al., 2015), we use the full-length F1 variant of Rouge (Lin, 2004) for the Gigaword results, and the 75-bytes length limited Recall variant of Rouge for DUC. Additionally, we also report other standard language generation metrics (as motivated recently by See et al. (2017)): METEOR (Denkowski and Lavie, 2014), BLEU-4 (Papineni et al., 2002), and CIDEr-D (Vedantam et al., 2015), based on the MS-COCO evaluation script (Chen et al., 2015).

4.3 Training Details

We use the following simple settings for all the models, unless otherwise specified. We unroll the encoder RNN's to a maximum of 50 time steps and decoder RNN's to a maximum of 30 time steps.

[1] http://duc.nist.gov/duc2004/tasks.html

Models	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU-4	CIDEr-D
PREVIOUS WORK						
ABS+ (Rush et al., 2015)	29.76	11.88	26.96	-	-	-
RAS-Elman (Chopra et al., 2016)	33.78	15.97	31.15	-	-	-
words-lvt2k-1sent (Nallapati et al., 2016)	32.67	15.59	30.64	-	-	-
OUR MODELS						
Baseline	31.75	14.71	29.91	14.54	10.31	128.22
Multi-Task with Entailment Generation	32.75	15.35	30.82	15.25	11.09	130.44

Table 1: Summarization results on Gigaword. Rouge scores are full length F-1, following previous work.

We use RNN hidden state dimension of 512 and word embedding dimension of 256. We do not initialize our word embeddings with any pre-trained models, i.e., we learn them from scratch. We use the Adam (Kingma and Ba, 2015) optimizer with a learning rate of 0.001. During training, to handle the large vocabulary, we use the sampled loss trick of Jean et al. (2014). We always tune hyperparameters on the validation set of the corresponding dataset, where applicable. For multi-task learning, we tried a few mixing ratios and found 1 : 0.05 to work better, i.e., 100 mini-batches of summarization with 5 mini-batches of entailment generation task in alternate training rounds.

5 Results and Analysis

5.1 Summarization Results: Gigaword

Baseline Results and Previous Work Our baseline is a strong encoder-attention-decoder model based on Luong et al. (2015) and presented by Chopra et al. (2016). As shown in Table 1, it is reasonably close to some of the state-of-the-art (comparable) results in previous work, though making this baseline further strong (e.g., based on pointer-copy mechanism) is our next step.

Multi-Task Results We show promising initial multi-task improvements on top of our baseline, based on several metrics. This suggests that the entailment generation model is teaching the summarization model some skills about how to choose a logical subset of the events in the full input document. This is especially promising given that the domain of the entailment dataset (image captions) is very different from the domain of the summarization datasets (news), suggesting that the model might be learning some domain-agnostic inference skills.

5.2 Summarization Results: DUC

Here, we directly use the Gigaword-trained model to test on the DUC-2004 dataset (see tuning discussion in Sec. 4.1). In Table 2, we again see that our Luong et al. (2015) baseline model achieves competitive performance with previous work, esp. on Rouge-2 and Rouge-L. Next, we show promising multi-task improvements over this baseline of around 0.4% across all metrics, despite being a test-only setting and also with the mismatch between the summarization and entailment domains.

Models	R-1	R-2	R-L
Rush et al. (2015)	28.18	8.49	23.81
Chopra et al. (2016)	28.97	8.26	24.06
Nallapati et al. (2016)	28.35	9.46	24.59
Baseline	27.74	8.82	24.45
Multi-Task	28.17	9.22	24.84

Table 2: Summarization test results on DUC-2004 corpus. Rouge scores are based on 75-byte Recall, following previous work.

Input Document: *results from the second round of the french first-division soccer league -lrb- home teams listed first -rrb- : UNK*
Ground-truth Summary: *french soccer results*
Baseline Summary: *first round results of french league soccer league*
Multi-task Summary: *second round of french soccer league results*

Input Document: *austrian women in leading positions complained about lingering male domination in their society in a meeting tuesday with visiting u.s. first lady hillary rodham clinton .*
Ground-truth Summary: *austrian women complain to mrs. clinton about male domination by roland prinz*
Baseline Summary: *first lady meets with first lady*
Multi-task Summary: *austrian women complained about male domination*

Figure 3: Output examples of our multi-task model in comparison with the baseline.

5.3 Analysis Examples

Figure 3 shows some additional interesting output examples of our multi-task model and how it generates summaries that are better at being logically entailed by the input document, whereas the baseline model contains some crucial contradictory or unrelated information.

6 Conclusion and Next Steps

We presented a multi-task learning approach to incorporate entailment generation knowledge into summarization models. We demonstrated promising initial improvements based on multiple datasets and metrics, even when the entailment knowledge was extracted from a domain different from the summarization domain.

Our next steps to this workshop paper include: (1) stronger summarization baselines, e.g., using pointer copy mechanism (See et al., 2017; Nallapati et al., 2016), and also adding this capability to the entailment generation model; (2) results on CNN/Daily Mail corpora (Nallapati et al., 2016); (3) incorporating entailment knowledge from other news-style domains such as the new Multi-NLI corpus (Williams et al., 2017), and (4) demonstrating mutual improvements on the entailment generation task.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work was supported by a Google Faculty Research Award, an IBM Faculty Award, a Bloomberg Data Science Research Grant, and NVidia GPU awards.

References

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

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*.

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*.

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

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.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *CoRR*.

Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *ANLP*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

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

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.

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

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, pages 1412–1421.

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.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

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

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

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

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575.

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