Robust Neural Abstractive Summarization Systems and Evaluation against Adversarial Information

Lisa Fan[1] **Dong Yu**[2] **Lu Wang**[1]
[1]College of Computer and Information Science, Northeastern University
[2]Tencent AI Lab
[1]lisafan@ccs.neu.edu, luwang@ccs.neu.edu
[2]dyu@tencent.com

Abstract

Sequence-to-sequence (seq2seq) neural models have been actively investigated for abstractive summarization. Nevertheless, existing neural abstractive systems frequently generate factually incorrect summaries and are vulnerable to adversarial information, suggesting a crucial lack of semantic understanding. In this paper, we propose a novel semantic-aware neural abstractive summarization model that learns to generate high quality summaries through semantic interpretation over salient content. A novel evaluation scheme with adversarial samples is introduced to measure how well a model identifies off-topic information, where our model yields significantly better performance than the popular pointer-generator summarizer. Human evaluation also confirms that our system summaries are uniformly more informative and faithful as well as less redundant than the seq2seq model.

1 Introduction

Automatic text summarization holds the promise of alleviating the information overload problem (Jones et al., 1999). Considerable progress has been made over decades, but existing summarization systems are still largely extractive—important sentences or phrases are identified from the original text for inclusion in the output (Nenkova et al., 2011). Extractive summaries thus unavoidably suffer from redundancy and incoherence, leading to the need for abstractive summarization methods. Built on the success of sequence-to-sequence (seq2seq) learning models (Sutskever et al., 2014), there has been a growing interest in utilizing a neural framework for abstractive summarization (Rush et al., 2015; Nallapati et al., 2016; Wang and Ling, 2016; Tan et al., 2017; Chen and Bansal, 2018).

Although current state-of-the-art neural models naturally excel at generating grammatically correct sentences, the model structure and learning objectives have intrinsic difficulty in acquiring semantic interpretation of the input text, which is crucial for summarization. Importantly, the lack of semantic understanding causes existing systems to produce *unfaithful generations*. Cao et al. (2018) report that about 30% of the summaries generated from a seq2seq model contain fabricated or nonsensical information.

Furthermore, current neural summarization systems can be easily fooled by off-topic information. For instance, Figure 1 shows one example where irrelevant sentences are added into an article about "David Collenette's resignation". Both the seq2seq attentional model (Nallapati et al., 2016) and the popular pointer-generator model (See et al., 2017) are particularly susceptible to unfaithful generation, partially because these models tend to rely on sentences at the beginning of the articles for summarization while being ignorant about their content. Therefore, we design a novel *adversarial evaluation* metric to measure the robustness of each summarizer against small amounts of randomly inserted topic-irrelevant information. The intuition is that if a summarization system truly understands the salient entities and events, it would ignore unrelated content.

Article Snippet: *For years Joe DiMaggio was always introduced at Yankee Stadium as "baseball's greatest living player." But with his memory joining those of Babe Ruth, Lou Gehrig, Mickey Mantle and Miller Huggins.* Canada's Minister of Defense resigned today, a day after an army official testified that top military officials had altered documents to cover up responsibility for the beating death of a Somali teen-ager at the hands of Canadian peacekeeping troops in 1993. Defense minister David Collenette insisted that his resignation had nothing to do with the Somalia scandal. *Ted Williams was the first name to come to mind, and he's the greatest living hitter. ...*

Seq2seq: George Vecsey sports of The Times column on New York State's naming of late baseball legend Joe DiMaggio as "baseball's greatest living player," but with his memory joining those of Babe Ruth, Lou Gehrig, Mickey Mantle and Miller dens.
Pointer-generator: Joe DiMaggio is first name to come to mind, and Ted Williams is first name to come to mind, and he's greatest living hitter; he will be replaced by human resources minister, Doug Young, and will keep his Parliament seat for governing Liberal Party.
Our Model: Former Canadian Defense Min David Collenette resigns day after army official testifies that top military officials altered documents to cover up responsibility for beating death of Somali teen-ager at hands of Canadian peacekeeping troops in 1993.

Figure 1: Sample summaries for an adversarial example. The inserted off-topic sentences are in *italics*. The remainder of the input, truncated for space, consists entirely of the David Collenette story.

To address the above issues, we propose a novel *semantic-aware abstractive summarization model*, inspired by the human process of writing summaries—important events and entities are first identified, and then used for summary construction. Concretely, taking an article as input, our model first generates a set of summary-worthy semantic structures consisting of predicates and corresponding arguments (as in semantic parsing), then constructs a fluent summary reflecting the semantic information. Both tasks are learned under an encoder-decoder architecture with new learning objectives. A dual attention mechanism for summary decoding is designed to consider information from both the input article and the generated predicate-argument structures. We further present a novel decoder with a segment-based reranking strategy to produce diverse hypotheses and reduce redundancy under the guidance of generated semantic information.

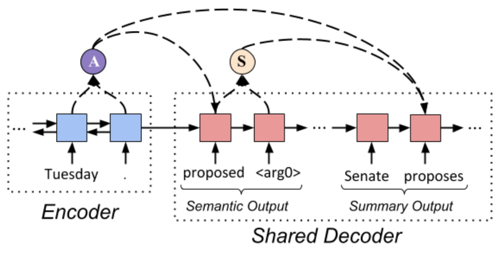
Evaluation against adversarial samples shows that while performance by the seq2seq attentional model and the pointer-generator model is impacted severely by even a small addition of topic-irrelevant information to the input, our model is significantly more robust and consistently produces more on-topic summaries (i.e. higher ROUGE and METEOR scores for standard automatic evaluation). Our model also achieves significantly better ROUGE and METEOR scores than both models on the benchmark dataset CNN/Daily Mail (Hermann et al., 2015). Specifically, our model's summaries use substantially fewer and shorter extractive fragments than the comparisons and have less redundancy, alleviating another common problem for the seq2seq framework. Human evaluation demonstrates that our model generates more informative and faithful summaries than the seq2seq model.

2 Related Work

To discourage the generation of fabricated content in neural abstractive models, a pointer-generator summarizer (See et al., 2017) is proposed to directly reuse words from the input article via a copying mechanism (Gu et al., 2016). However, as reported in their work (See et al., 2017) and confirmed by our experiments, this model produces nearly extractive summaries. While maintaining their model's rephrasing ability, here we improve the faithfulness and informativeness of neural summarization models by enforcing the generation of salient semantic structures via multi-task learning and a reranking-based decoder.

Our work is in part inspired by prior abstractive summarization work, where the summary generation process consists of a distinct content selection step (i.e., what to say) and a surface realization step (i.e., how to say it) (Wang and Cardie, 2013; Pighin et al., 2014). Our model learns to generate salient semantic roles and a summary in a single end-to-end trained neural network.

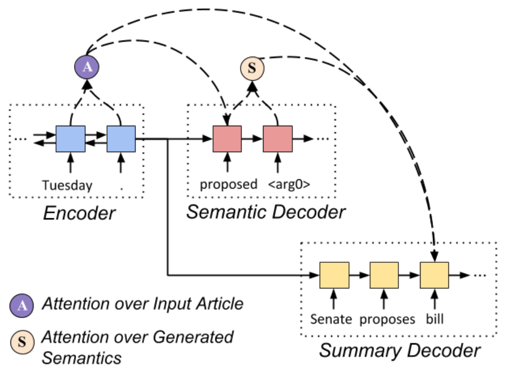
Our proposed model also leverages the recent successes of multi-task learning (MTL) as applied to neural networks for a wide array of natural language processing tasks (Luong et al., 2015; Søgaard and Goldberg, 2016; Peng et al., 2017; Rei, 2017; Isonuma et al., 2017). Most recent work (Pasunuru et al., 2017) leverages MTL to jointly improve performance on summarization and entailment generation. Instead of treating the tasks equally, we employ semantic parsing for the sake of facilitating more informative and faithful summary generation.



(a) Shared Decoder Model

Input:	
Article:	the Senate proposed a tax bill on Tuesday .
Output:	
Semantic:	proposed <ARG0> the Senate <ARG1> tax bill
Summary:	Senate proposes bill .

(b) Sample Input and System Outputs



(c) Separate Decoder Model

Figure 2: Our semantic-aware summarization model with dual attention over both input and generated salient semantic roles. In both shared and separate decoder models, the semantic roles are generated first, followed by the summary construction. Best viewed in color.

3 Model

3.1 Model Formulation

In the standard seq2seq model, a sequence of input tokens $\mathbf{x} = \{x_1, ..., x_n\}$ is encoded by a recurrent neural network (RNN) with hidden states $\mathbf{h}_i$ at every timestep i. The final hidden state of the encoder is treated as the initial state of the decoder, another RNN ($\mathbf{h}_n = \mathbf{s}_0$). The decoder will produce the output sequence $\mathbf{y} = \{y_1, ..., y_T\}$, with hidden states $\mathbf{s_t}$ at every timestep t. Our model outputs two sequences: the sequence of semantic role tokens $\mathbf{y}^s = \{y_1^s, ..., y_{T^s}^s\}$ is generated first, followed by the sequence of summary (or abstract) tokens $\mathbf{y}^a = \{y_1^a, ..., y_{T^a}^a\}$.

The attention mechanism of Bahdanau et al. (2014) is utilized to attend the input. Concretely, a context vector $\mathbf{c}_t^{\text{inp}}$ is calculated as the summation of the encoder hidden states, weighted by the attention distribution $\mathbf{a}_t$ at every timestep t:

$$\mathbf{c}_t^{\text{inp}} = \sum_{i=1}^{n} a_{it}\mathbf{h}_i \tag{1}$$

$$\mathbf{a}_t = \text{softmax}(\mathbf{e}_t) \tag{2}$$

$$e_{it} = \mathbf{v}\tanh(\text{W}_s\mathbf{s}_t + \text{W}_h\mathbf{h}_i) \tag{3}$$

where $\mathbf{v}, \text{W}_s, \text{W}_h$ are learnable parameters. The context vector, along with the decoder hidden state, is used to produce the vocabulary distribution via softmax:

$$p(y_t \mid x_t, \mathbf{s}_t, \mathbf{c}_t^{inp}) = \text{softmax}(\mathbf{o}_t) \tag{4}$$

$$\mathbf{o}_t = \text{W}_v[\mathbf{s}_t \,\|\, \mathbf{c}_t^{\text{inp}}] + \mathbf{d} \tag{5}$$

where W_v and $\mathbf{d}$ are learned parameters and $\|$ represents vector concatenation.

The loss is defined as the negative log likelihood of generating summary $\mathbf{y}$ over the training set D using model parameters θ:

$$\text{loss} = \sum_{(\mathbf{y},\mathbf{x})\in D} -\log p(\mathbf{y} \mid \mathbf{x}; \theta) \tag{6}$$

The log probability for each training sample is the average log likelihood across decoder timesteps.

Encoder. In our models, our encoder is a single-layer bidirectional long short-term memory (LSTM) unit (Graves and Schmidhuber, 2005), where the hidden state is a concatenation of the forwards and backwards LSTMs: $\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i} \,\|\, \overleftarrow{\mathbf{h}_i}]$.

Decoders. We propose two different decoder architectures—*separate decoder* and *shared decoder*—to handle semantic information. In the separate decoder model (See Figure 2c), the *semantic decoder* and the *summary decoder* are each implemented as their own single-layer LSTM. This setup, inspired by the one-to-many multi-task learning framework of Luong et al. (2015), encourages each decoder to focus more on its respective task. Decoder output and attention over the input are calculated using Eqs. 1-5, but separate sets of parameters are learned for each decoder: $\mathbf{v}^s, \mathrm{W}_s^s, \mathrm{W}_h^s, \mathrm{W}_v^s, \mathbf{d}^s$ for the semantic decoder, and $\mathbf{v}^a, \mathrm{W}_s^a, \mathrm{W}_h^a, \mathrm{W}_v^a, \mathbf{d}^a$ for the summary decoder.

We further study a shared decoder model (See Figure 2a) for the purpose of reducing the number of parameters as well as increasing the summary decoder's exposure to semantic information. One single-layer LSTM is employed to sequentially produce the important semantic structures, followed by the summary. Our output thus becomes $\mathbf{y} = [\mathbf{y}_s \parallel \mathbf{y}_a]$, and the first timestep of the summary decoder is the last timestep of the semantic decoder. Attention is calculated as in Eqs. 1-5.

For both models, the loss becomes the weighted sum of the semantic loss and the summary loss:

$$\text{loss} = -\sum_{(\mathbf{y},\mathbf{x})\in D} \alpha \log p(\mathbf{y}^a \mid \mathbf{x};\theta) + (1-\alpha)\log p(\mathbf{y}^s \mid \mathbf{x};\theta) \tag{7}$$

In our experiments, we set α as 0.5 unless otherwise specified.

We also investigate *multi-head attention* (Vaswani et al., 2017) over the input to acquire different language features. To our knowledge, we are the first to apply it to the task of summarization. As shown later in the results section, this method is indeed useful for summarization, with different heads learning different features. In fact, the multi-head attention is particularly well-suited for our shared decoder model, as some heads learn to attend semantic aspects and others learn to attend summary aspects.

Target Semantic Output. We use semantic role labeling (SRL) as the target semantic representation, which identifies predicate-argument structures. To create the target data, articles in the training set are parsed in the style of PropBank (Kingsbury and Palmer, 2002) using the DeepSRL parser from He et al. (2017). We then choose up to five SRL structures that have the most overlap with the reference summary. Here we first consider matching headwords of predicates and arguments. If no match is found, we consider all content words. Note that the semantic labels are only used for model training. At test time, no external resource beyond the article itself is used for summary decoding.

3.2 Dual Attention over Input and Semantics

To further leverage semantic information, a dual attention mechanism is used to attend over the generated semantic output in addition to the input article during summary decoding. Although attending over multiple sources of information has been studied for visual QA (Nam et al., 2017) and sentence-level summarization (Cao et al., 2018), these are computed over different *encoder states*. In contrast, our dual attention mechanism considers *decoding results* from the semantic output. Although our attention mechanism may appear close to intra- or self-attention (Sankaran et al., 2016), the function of our dual attention is to attend over a specific portion of the previously generated content that represents its own body of information, whileas traditional self-attention is predominantly used to discourage redundancy.

The context vector attending over the semantic decoder hidden states is calculated as follows:

$$\mathbf{c}_{t'}^{\text{sem}} = \sum_{j=1}^{T^s} b_{jt'}\mathbf{s}_j^s \tag{8}$$

$$\mathbf{b}_{t'} = \text{softmax}(\mathbf{f}_{t'}) \tag{9}$$

$$f_{jt'} = \mathbf{v}' \tanh(\mathrm{W}'_s \mathbf{s}_{t'}^a + \mathrm{W}'_h \mathbf{s}_j^s) \tag{10}$$

where $\mathbf{s}^a$ are the summary decoder's hidden states, $\mathbf{s}^s$ are the semantic decoder's hidden states, and t' are the time steps for the summary decoder. The summary output $\mathbf{o}_{t'}^a$ now becomes:

$$\mathbf{o}_{t'}^a = \mathrm{W}'_v[\mathbf{s}_{t'}^a \parallel \mathbf{c}_{t'}^{\text{inp}} \parallel \mathbf{c}_{t'}^{\text{sem}}] + \mathbf{d}' \tag{11}$$

3.3 Reranking-based Summary Decoder

Albeit a standard practice, the beam search algorithm has been shown to produce suboptimal results in neural text generation problems (Vijayakumar et al., 2016) due to the fact that one beam often dominates others, yielding hypotheses that only differ by the last token. To combat this issue, others have proposed beam rerankers that utilize external features such as Maximum Mutual Information (Li et al., 2016), segment-reranking (Shao et al., 2017), and hamming loss (Wen et al., 2015). Here, we discuss our own segment-reranking beam search decoder, which encourages both *beam diversity* and *semantic coverage*. Our reranker is applied only during summary decoding, where we rely on the generated semantic roles for global guidance. The only modification made to the semantic decoder is a de-duplication strategy, where generated predicates seen in previous output are eliminated.

Reranking. Regular beam search chooses the hypotheses for the next timestep solely based on conditional likelihood (i.e., $p(\mathbf{y}^a \mid \mathbf{x})$). In our proposed summary decoder, we also *leverage our generated semantic information* and *curb redundancy* during beam selection. Reranking is performed every R timesteps based on the following scorer, where hypotheses with less repetition (r) and covering more content words from the generated semantics (s) are ranked higher:

$$\text{score} = \log\left(p(\mathbf{y}^a \mid \mathbf{x})\right) + \alpha r + \beta s \tag{12}$$

$$r = \log\left(1 - \frac{\text{\# LRS} * |\text{LRS}|}{\text{\# tokens in } \mathbf{y}^a}\right) \tag{13}$$

where LRS is the longest repeating substring in the current hypotheses. We define a repeating substring as a sequence of three or more tokens that appears in that order more than once in the hypothesis, with the intuition that long repeating fragments (ex. "`the Senate proposed a tax bill; the Senate proposed a tax bill.`") should be penalized more heavily than short ones (ex. "`the Senate proposed a tax bill; Senate proposed.`"). s measures the percentage of generated semantic words reused by the current summary hypothesis, contingent on the predicate of semantic structure matching. At every other timestep, we rank the hypotheses based on conditional likelihood and a weaker redundancy handler, r', that considers unigram novelty (i.e., percentage of unique content words): $\text{score} = \log\left(p(\mathbf{y}^a \mid \mathbf{x})\right) + \alpha' r'$. We use $R = 10$, $\alpha = 0.4$, $\beta = 0.1$, $\alpha' = 0.1$ in our experiments.

Beam Diversity. In the standard beam search algorithm, all possible extensions to the beams are considered, resulting in a comparison of $B \times \mathcal{D}$ hypotheses, where B is the number of beams and $\mathcal{D}$ is the vocabulary size. In order to make the best use of the reranking algorithm, we develop two methods to enforce hypothesis diversity during non-reranking timesteps.

Beam Expansion. Inspired by Shao et al. (2017), we rank only the K highest scoring extensions from each beam, where $K < B$. This ensures that at least two unique hypotheses from the previous timestep will carry on to the next timestep.

Beam Selection. We further reinforce hypothesis diversity through the two-step method of (1) likelihood selection and (2) dissimilarity selection. In likelihood selection, we accept N hypotheses (where $N < B$) from our pool of $B \times K$ based solely on conditional probabilities, as in traditional beam search. From the remaining hypotheses pool, we select $B - N$ hypotheses on the basis of dissimilarity. We choose the hypotheses with the highest dissimilarity score $\Delta([h]_N, h')$, where h' is a candidate and $[h]_N$ are the hypotheses chosen during likelihood selection. In our experiments, we use token-level Levenshtein edit distance (Levenshtein, 1966) as the dissimilarity metric, where $\Delta([h]_N, h') = \max_{h \in [h]_N} (\text{Lev}(h, h'))$. In experiments, we use $B = 12$, $K = 6$, $N = 6$.

4 Experimental Setup

Datasets. We experiment with two popular large datasets of news articles paired with human-written summaries: the CNN/Daily Mail corpus (Hermann et al., 2015) (henceforth CNN/DM) and the New York Times corpus (Sandhaus, 2008) (henceforth NYT). For CNN/DM, we follow the experimental setup from See et al. (2017) and obtain a dataset consisting of 287,226 training pairs, 13,368 validation pairs, and 11,490 test pairs. For NYT, we removed samples with articles of less than 100 words or summaries of less than 20 words. We further remove samples with summaries containing information outside the article, e.g., "[AUTHOR]'s movie review on..." where the author's name does not appear in the article. NYT consists of 280,146 training pairs and 15,564 pairs each for validation and test.

Training Details and Parameters. For all experiments, a vocabulary of 50k words shared by input and output is used. Model parameters and learning rate are adopted from prior work (See et al., 2017) for comparison purpose. All models are also trained in stages of increasing maximum token lengths to expedite training. The models trained on the NYT dataset use an additional final training stage, where we optimized only on the summary loss (i.e., $\alpha = 1$ in Eq. 7). During decoding, unknown tokens are replaced with the highest scoring word in the corresponding attention distribution.

Baselines and Comparisons. We include as our extractive baselines TEXTRANK (Mihalcea and Tarau, 2004) and LEAD-2, the first 2 sentences of the input article, simulating the average length of the target summaries. We consider as abstractive comparison models (1) vanilla seq2seq with attention (SEQ2SEQ) and (2) pointer-generator (See et al., 2017) (POINT-GEN), which is trained from scratch using the released code. Results for variants of our model are also reported.[1]

Automatic Evaluation Metrics. For automatic evaluation, we first report the F1 scores of ROUGE-1, 2, and L (Lin and Hovy, 2003), and METEOR scores based on exact matching and full matching that considers paraphrases, synonyms, and stemming (Denkowski and Lavie, 2014).

We further measure two important aspects of summary quality: *extractiveness*—how much the summary reuses article content verbatim, and *redundancy*—how much the summary repeats itself. For the first aspect, we utilize the density metric proposed by Grusky et al. (2018) that calculates "the average length of extractive fragments":

$$\textsc{Density}(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} |f|^2 \tag{14}$$

where A represents the article, S represents the summary, and $F(A, S)$ is a set of greedily matched extractive fragments from the article-summary pair.

Based on density, we propose a new redundancy metric:

$$\textsc{Redundancy}(S) = \frac{1}{|S|} \sum_{f' \in F'(S)} (\#f' \times |f'|)^2 \tag{15}$$

where $F'(S)$ contains a set of fragments at least three tokens long that are repeated within the summary, and $\#f'$ is the repetition frequency for fragment f'. Intuitively, longer fragments and more frequent repetition should be penalized more heavily.

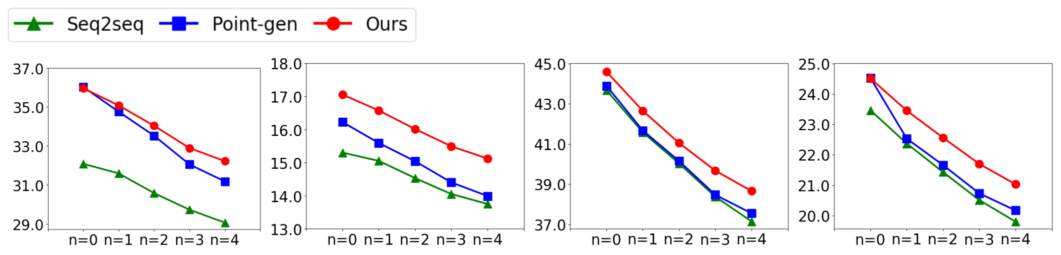
5 Results

Adversarial Evaluation. Our pilot study suggests that the presence of minor irrelevant details in a summary often hurts a reader's understanding severely, but such an error cannot be captured or penalized by the recall-based ROUGE and METEOR metrics.

In order to test a model's ability to discern irrelevant information, we design a novel adversarial evaluation scheme, where we purposely mix a limited number of off-topic sentences into a test article. The intuition is that if a summarization system truly understands the salient entities and events, it would ignore unrelated sentences.

We randomly select 5,000 articles from the CNN/DM test set with the "news" tag in the URL (mainly covering international or domestic events). For each article, we randomly insert one to four sentences from articles in the test set with the "sports" tag. For NYT, we randomly select 5,000 articles from the test set with government related tags ("U.S.", "Washington", "world") and insert one to four sentences from articles outside the domain ("arts", "sports", "technology"). We ensure that sentences containing a pronoun in the first five words are not interrupted from the previous sentence so that discourse chains are unlikely to be broken. the content of the article, it would ignore unrelated sentences. We took 15,000 articles from the NYT corpus with government related tags ("u.s.", "washington", "world"), and for each article, randomly inserted up to four sentences from articles outside the domain ("arts", "sports", "technology"). We ensured sentences that contained a pronoun in the first five words were not interrupted from the previous sentence so as not to break discourse or coreference chains.

[1]Reinforcement learning methods have been recently proposed for abstractive summarization (Paulus et al.; Celikyilmaz et al., 2018; Chen and Bansal, 2018). We will study this type of learning models in future work.



(b) CNN/DM ROUGE-L (c) CNN/DM METEOR (d) NYT ROUGE-L (e) NYT METEOR

Figure 3: ROUGE-L and full METEOR scores on adversarial samples, where n irrelevant sentences are inserted into original test articles. Our models (shared decoder for NYT, shared+MHA for CNN/DM) are sturdier against irrelevant information than seq2seq and pointer-generator.

	CNN/DailyMail							NYT						
	ROUGE			**METEOR**		**Dens.**	**Red.**	**ROUGE**			**METEOR**		**Dens.**	**Red.**
	R-1	R-2	R-L	exact	full			R-1	R-2	R-L	exact	full		
HUMAN	-	-	-	-	-	3.40	0.06	-	-	-	-	-	4.32	0.12
LEAD-2	38.35	15.74	31.48	14.44	15.62	59.19	0.33	39.01	21.96	30.84	16.73	17.91	68.31	0.12
TEXTRANK	33.78	12.41	28.75	12.73	13.94	36.44	7.94	34.81	16.02	25.56	14.69	16.14	49.11	0.65
Abstractive Comparisons														
SEQ2SEQ	32.14	12.33	29.41	13.66	14.65	8.46	11.92	44.53	28.54	37.05	19.62	20.54	7.31	3.09
POINT-GEN	35.60	15.24	32.43	15.50	16.54	16.80	10.77	46.76	31.13	38.85	20.62	21.55	9.54	2.67
Our Semantic-Aware Models														
DEC$_{share}$	35.44	14.18	32.28	14.80	15.84	12.42	5.97	**46.40**	**30.55**	**38.47**	**19.91**	**20.79**	8.73	**0.83**
+ MHA	**36.31**	14.96	**33.21**	**15.56**	**16.62**	13.03	**5.20**	45.88	30.26	38.23	19.32	20.17	**8.33**	1.29
DEC$_{sep}$	35.31	13.97	32.23	14.81	15.86	**11.55**	5.21	45.98	30.23	38.27	19.47	20.33	8.68	1.10
+ MHA	36.07	**15.09**	33.08	15.25	16.29	12.93	5.65	46.10	30.37	38.41	19.43	20.29	8.58	0.89

Table 1: Results on CNN/Daily Mail and NYT. For our model, we display four variants, based on shared or separate decoder, and with or without multi-head attention (MHA). In addition to ROUGE and METEOR, we also display extractive density (Dens.) and redundancy (Red.) (lower scores are preferred). Best performing amongst our models are in **bold**. All our ROUGE scores have a 95% confidence interval of at most ±0.04. Our models all statistically significantly outperform seq2seq for ROUGE and METEOR (approximate randomization test, $p < 0.01$). Our shared decoder with MHA model statistically significantly outperforms pointer-generator in ROUGE-1 and ROUGE-L on CNN/DM.

The adversarial evaluation results on seq2seq, pointer-generator, and our shared decoder model are shown in Figure 3, where our model consistently yields significantly better ROUGE-L and METEOR scores than the comparisons, and is less affected as more irrelevant sentences are added. Sample summaries for an adversarial sample are shown in Figure 1. We find that our semantic decoder plays an important role in capturing salient predicates and arguments, leading to higher quality summaries.

Automatic Evaluation. The main results are displayed in Table 1. On CNN/DM, all of our models significantly outperform seq2seq across all metrics, and our shared decoder model with multi-head attention yields significantly better ROUGE (R-1 and R-L) scores than the pointer-generator on the same dataset (approximate randomization test, $p < 0.01$). Despite the fact that both ROUGE and METEOR favor recall and thus reward longer summaries, our summaries that are often shorter than comparisons still produce significantly better ROUGE and METEOR scores than seq2seq on the NYT dataset.

Furthermore, our system summaries, by all model variations, *reuse fewer and shorter phrases from the input* (i.e., lower density scores) than the pointer-generator model, signifying a potentially stronger ability to rephrase. Note that the density scores for seq2seq are likely deflated due to its inability to handle out-of-vocabulary words. Our models also produce *less redundant* summaries than their abstractive comparisons.

Human Evaluation. We further conducted a pilot human evaluation on 60 samples selected from the NYT test set. For each article, the summaries by the seq2seq model, our shared decoder model, and the human reference were displayed in a randomized order. Two human judges, who are native or fluent English speakers, were asked to read the article and rank the summaries against each other based on non-redundancy, fluency, faithfulness to input, and informativeness (whether the summary delivers the main points of the article). Ties between system outputs were allowed but discouraged.

As illustrated in Table 2, our model was ranked significantly higher than seq2seq across all metrics. Surprisingly, our model's output was ranked higher than the reference summary in 26% of the samples for informativeness and fluency. We believe this is due to the specific style of the NYT reference summaries and the shorter lengths of our summaries: the informal style of the reference summaries (e.g., frequently dropping subjects and articles) negatively affects its fluency rating, and readers find our shorter summaries to be more concise and to the point.

	NON-RED.	FLUENCY	FAITH.	INFORM.
HUMAN	1.4 ± 0.7	1.5 ± 0.7	1.5 ± 0.8	1.6 ± 0.8
SEQ2SEQ	2.0 ± 0.8	2.4 ± 0.7	2.1 ± 0.8	2.4 ± 0.7
OURS	1.6 ± 0.7	2.1 ± 0.8	1.9 ± 0.7	2.0 ± 0.7

Table 2: Human ranking results on non-redundancy, fluency, faithfulness of summaries, and informativeness. The mean ($\pm$ std. dev.) for the rankings is shown (lower is better). Across all metrics, the difference between our model (shared decoder) and human summary, as well as between our model and seq2seq is statistically significant (one-way ANOVA, $p < 0.05$).

6 Discussion

Usage of Semantic Roles in Summaries. We examine the utility of the generated semantic roles. Across all models, approximately 44% of the generated predicates are part of the reference summary, indicating the adequacy of our semantic decoder. Furthermore, across all models, approximately 65% of the generated predicates are reused by the generated summary, and approximately 53% of the SRL structures are reused by the system using a strict matching constraint, in which the predicate and head words for all arguments must match in the summary. When gold-standard semantic roles are used for dual attention in place of our system generations, ROUGE scores increase by about half a point, indicating that improving semantic decoder in future work will further enhance the summaries.

Coverage. We also conduct experiments using a coverage mechanism similar to the one used in See et al. (2017). We apply our coverage in two places: (1) over the input to handle redundancy, and (2) over the generated semantics to promote its reuse in the summary. However, no significant difference is observed. Our proposed reranker handles both issues in a more explicit way, and does not require the additional training time used to learn coverage parameters.

Alternative Semantic Representation. Our summarization model can be trained with other types of semantic information. For example, in addition to using the salient semantic roles from the input article, we also explore using SRL parses of the reference abstracts as training signals, but the higher level of abstraction required for semantic generation hurts performance by two ROUGE points for almost all models, indicating the type of semantic structure matters greatly for the ultimate summarization task.

For future work, other semantic representation along with novel model architecture will be explored. For instance, other forms of semantic representation can be considered, such as frame semantics (Baker et al., 1998) or Abstract Meaning Representation (AMR) (Banarescu et al., 2013). Although previous work by Vinyals et al. (2015) has shown that seq2seq models are able to successfully generate linearized tree structures, we may also consider generating semantic roles with a hierarchical semantic decoder (Sordoni et al., 2015).

7 Conclusion

We presented a novel semantic-aware neural abstractive summarization model that jointly learns summarization and semantic parsing. A novel dual attention mechanism was designed to better capture the semantic information for summarization. A reranking-based decoder was proposed to promote the content coverage. Our proposed adversarial evaluation demonstrated that our model was more adept at handling irrelevant information compared to popular neural summarization models. Experiments on two large-scale news corpora showed that our model yielded significantly more informative, less redundant, and less extractive summaries. Human evaluation further confirmed that our summaries were more informative and faithful than comparisons.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1662–1675.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of ACL*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1631–1640.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*.

Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. 2017. Extractive summarization using multi-task learning with document classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2101–2110, Copenhagen, Denmark. Association for Computational Linguistics.

K Sparck Jones et al. 1999. Automatic summarizing: factors and directions. *Advances in automatic text summarization*, pages 1–12.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*.

VI Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, page 707.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 71–78.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Ça glar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *CoNLL 2016*, page 280.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 299–307.

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.

Ramakanth Pasunuru, Han Guo, and Mohit Bansal. 2017. Towards improving abstractive summarization via entailment generation. In *Proceedings of the Workshop on New Frontiers in Summarization*.

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

Hao Peng, Sam Thomson, and Noah A Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1.

Daniele Pighin, Marco Cornolti, Enrique Alfonseca, and Katja Filippova. 2014. Modelling events through memory-based, open-ie patterns for abstractive summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–901, Baltimore, Maryland. Association for Computational Linguistics.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130, Vancouver, Canada. Association for Computational Linguistics.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Baskaran Sankaran, Haitao Mi, Yaser Al-Onaizan, and Abe Ittycheriah. 2016. Temporal attention model for neural machine translation. *arXiv preprint arXiv:1608.02927*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219, Copenhagen, Denmark. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–235.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562. ACM.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria. Association for Computational Linguistics.

Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 47–57, San Diego, California. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Prague, Czech Republic. Association for Computational Linguistics.

Supplementary Materials

A NYT preprocessing

We base our preprocessing method on those taken by Paulus et al. (2017). We first removed samples without a summary, as well as articles with less than 100 words and corresponding summaries of less than 20 words. The Stanford CoreNLP Tokenizer (Manning et al., 2014) is used to parse the articles and summaries. All words are made lowercase except those that are named entities (parsed with Stanford CoreNLP), and replace all numbers with "0". Template words at the ends of summaries are removed: "(s)", "(m)", "photo", "graph", "chart", "map", "table", and "drawing". We then removed samples where the summary contained the following in the first 5 words: "article", "column", "op-ed", "essay", "editorial", "letter", "profile", "interview", "excerpts", "news", "analysis", "review". These words were indicative of a template-style summary containing information outside articles, e.g., "[AUTHOR]'s movie review of..." where the author's name does not appear in the article. While this trimmed about 30% of our data, we found that our method may have been too conservative, as some column names longer than 5 words (ex. "Lisa Belkin life 's work column on ...") were not successfully omitted.

This gives us a dataset of 311,274 samples, where we put the samples in chronological publication order and took 280,146 samples (90%) for the training set, and 15,564 samples (5%) each for validation and test sets.

B Target Semantic Output Creation

We developed two algorithms for creating the target semantic outputs: strict matching and soft matching. Soft matching was only used for samples where the strict matching produced no results. In both cases, we first parse the article in the style of PropBank (Kingsbury and Palmer, 2002) using He et al.'s (2017) DeepSRL parser, and keep only the predicates, ARG0 (the agent), ARG1 (the patient), and ARG2 (instrument, benefactive, or attribute).

For the NYT dataset, 80% of the samples produced target SRL structures using strict matching, while only 0.01% of samples did not have any target SRL structures after the soft matching operation. Each sample had approximately 5 SRL structures per article.

Strict Matching. We first perform stemming on the output words using Stanford CoreNLP and consider only the last word in each argument. We define the head word to be the last word in the argument. We then perform the following matching algorithm for each SRL structure parsed from the article:

```
$pred$ = predicate
$Stop$ = [list of stop words]
$Summ$ = [list of summary words]
if $pred \in Stop$ or $pred \notin Summ$ then
    REJECT
else
    if $\exists$ ARG0 and ARG0 $\notin Stop$ then
        if ARG0 $\in Summ$ then
            ACCEPT
        else
            REJECT
    else if $\exists$ ARG1 and ARG1 $\notin Stop$ then
        if ARG1 $\in Summ$ then
            ACCEPT
        else
            REJECT
    else
        REJECT
```

Soft Matching. We first perform stemming on the output words using Stanford CoreNLP. We then count the number of unique non-stop words in the predicate, ARG0, and ARG1 that are also in the summary, and choose at most the top 5 SRL structures with at least two matched words in order of number of matched words.

C Multi-head Attention over Input

Multi-head attention (Vaswani et al., 2017) over the input has been used in various NLP tasks to acquire different language features. Concretely, K separate context vectors, or heads, attend over the input, and the final context vector $\mathbf{c}_t^{inp}$ takes the form of a linear transformation over the concatenation of all heads. Each context vector $\mathbf{c}_t^i$ is calculated using Eqs. 1-3, with independent parameters $\mathbf{v}^i$ and W_h^i. All attention heads share the same parameter W_s that transforms the decoder hidden state $\mathbf{s}_t$.

$$\mathbf{c}_t^i = \sum_{j}^{j=n} a_{jt}^i \mathbf{h}_j \tag{16}$$

$$\mathbf{a}_t^i = \mathrm{softmax}(\mathbf{e}_t^i) \tag{17}$$

$$e_{jt}^i = \mathbf{v}^i \tanh(\mathrm{W}_s \mathbf{s}_t + \mathrm{W}_h^i \mathbf{h}_j) \tag{18}$$

D Redundancy Handling

In our proposed beam search summary decoder, reranking is performed every R timesteps based on the following scorer, where hypotheses with less repetition (r) and covering more content words from the generated semantics (s) are ranked higher:

$$\mathrm{score} = \log\left(p(\mathbf{y}^a \,|\, \mathbf{x})\right) + \alpha r + \beta s \tag{19}$$

$$r = \log\left(1 - \frac{\text{\# LRS} * |\text{LRS}|}{\text{\# tokens in } \mathbf{y}^a}\right) \tag{20}$$

$$s = \frac{\text{\# unique argument tokens in } \mathbf{y}^s \text{ and } \mathbf{y}^a}{\text{\# unique tokens in } \mathbf{y}^s} \tag{21}$$

where LRS is the longest repeating substring in the current hypotheses. We define a repeating substring as a sequence of three or more tokens that appear more than once in the hypothesis, with the intuition that long extractive fragments should be penalized more heavily than short ones. Our longest repeating substring algorithm is implemented by finding the deepest non-leaf node in a suffix tree. At all other timesteps, we rank the hypotheses based on conditional likelihood and a weaker redundancy handler:

$$\mathrm{score} = \log\left(p(\mathbf{y}^a \,|\, \mathbf{x})\right) + \alpha' r' \tag{22}$$

$$r' = \frac{\text{\# unique tokens in } \mathbf{y}^a}{\text{\# tokens in } \mathbf{y}^a} \tag{23}$$

We omit stop words for r' and s.

E Finding Set of Repeating Fragments

We describe the procedure for finding the set of self-repeating substrings used in the redundancy automatic evaluation metric.
1. Extract all repeated substrings using a suffix tree, and only keep substrings ≥ 3 tokens.
2. Sort by number of occurrences, most to least.
3. Within the same number of occurrences, sort by substring length, longest to shortest.
4. For each substring in the sorted substrings, if the substring has no overlap with any items in results, add the substring to the results. Here, overlap is defined as "x is a substring of y" or "y is a substring of x".

The above procedure favors number of occurrences over length of repeated substring.

Article Snippet: Carmen Villegas did not expect the Roman Catholic Archdiocese of New York to send burly guards into her church, Our Lady Queen of Angels, but it did. She did not expect to be at the center of a chaotic scene, shouting at church officials and flinging open the front door of the church, but she was. She did not expect to be arrested, but she was, after she refused to leave. And she did not expect the archdiocese to close her beloved East Harlem church, two weeks early, but late Monday night, that is exactly what it did. Ms. Villegas and some fellow parishioners occupied the sanctuary for about 28 hours , protesting the archdiocese's plan to shut down Our Lady Queen of Angels on March 1. ...
Human Abstract: Roman Catholic Archdiocese of New York closes Our Lady Queen of Angels church in East Harlem two weeks earlier than planned after protests and vigil by parishioners lead to arrests.

Seq2seq: *Carmen [UNK], Roman Catholic Archdiocese of New York*, and five other pastor, Carmen [UNK], are evicted from church's East Harlem church after refusing to leave.
Our Model: Catholic Archdiocese of New York is closing its East Harlem church, two weeks early; Villegas and some fellow parishioners occupy sanctuary for about 28 hours, protesting archdiocese's plan to shut down Our Lady Queen of Angels on March 1.

Accompanied SRL Output:
<PRED> close <ARG0> the archdiocese <ARG1> her East Harlem church
<PRED> occupied <ARG0> Ms. Villegas and some fellow parishioners <ARG1> the sanctuary
<PRED> protesting <ARG0> Ms. Villegas and some fellow parishioners <ARG1> the archdiocese's plan to shut down Our Lady Queen of Angels on March 1

Figure 4: Sample summaries by our model and seq2seq. Errors are in *italics*. Our model generates salient semantic roles, which guide the construction of informative and correct summary. Semantic reusage by our summary is shown in color.

F Example Output

Figure 4 illustrates an example of our system summary. Our semantic decoder is able to generate structures that reuse arguments.

G Training Details

Experimental Setup. For all experiments, a vocabulary of 50k words shared by input and output is used. Word embeddings of size 128 are randomly initialized and learned during training. The single-head attention vector and hidden states of LSTMs used in all models have 256 dimensions. When multi-head attention is employed, four attention heads are learned, each of size 64. The Adagrad optimizer Duchi et al. (2011) is used with initial accumulator 0.1 and a learning rate of 0.15. We use a batch size of 16. The training process took 3 to 5 days depending on model and GPU type (both Quadro P5000 GPU and Tesla V100 GPU were used). Models all converge after 10 to 19 epochs.

Training in Stages. To expedite training time, we train all models in stages of increasing maximum token lengths—starting from highly truncated maximum lengths, we then increase the lengths when the models converge. Specifically, the three stages are of input/output token lengths 50/50, 200/50, and 400/100, where semantic and summary outputs each are truncated to the output token length. The models trained on the NYT dataset used a fourth training stage, in which we optimized only on the summary loss (ie. $\alpha = 1$ in Eqn 8).

Dealing with Unknown Tokens. In our models, we deal with the generated unknown tokens by replacing them with the highest scoring word in the corresponding attention distribution, since the generated unknown tokens very often correspond to out-of-vocabulary proper nouns. In multi-head attention models, we choose the highest scoring word in the summed attention distributions. This replacement method was shown to boost ROUGE scores by at least 1 point across all models.

		% generated / avg length		
	# SRL	arg0	arg1	arg2
reference	3.8	62.2 / 4.0	91.6 / 6.5	20.2 / 6.5
shared_dec	3.4	66.6 / 5.0	95.3 / 8.8	20.1 / 8.9

Table 3: Statistics on generated SRL.

Initializing with Seq2seq. We only include target SRL structures that appear in the system input. Therefore, for the highly truncated training stage, many training samples do not include a target semantic output. To ensure the model saw all the data, all semantic-aware models were initialized with the weights from the converged first stage baseline seq2seq model when possible ($\mathbf{v}^s, \mathrm{W}_s^s, \mathrm{W}_h^s, \mathrm{W}_v^s, \mathbf{d}^s$ are still initialized randomly for the separate decoder models). Both our original and modified decoders required the generated summary and semantic output to be at least 35 tokens long.

H Generated SRL Statistics

We found that our semantic decoder generates fewer SRL structures than the average target SRL parse, but generates a similar ratio of the individual arguments (See Table 3). The average length of our generated arguments is also longer than in the target SRL structures. In fact, the decoder will often first generate a structure with very long arguments, then generate subsequent structures from portions of the previous argument. For example, after generating ARG1 in the last SRL structure of Figure 4, the semantic decoder would likely next generate a structure with the predicate “shut” (as in “shut down”). This is partially a byproduct of imperfect target semantic parsings that exhibit a similar behavior. Improving the trade-off between argument lengths and the number of semantic structures (ie. generating more succinct semantic structures) is one direction to explore in future work.