Jointly Extracting and Compressing Documents with Summary State Representations

Citation for published version:

Digital Object Identifier (DOI):
10.18653/v1/n19-1397

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Jointly Extracting and Compressing Documents with Summary State Representations

Afonso Mendes ♠ Shashi Narayan♢∗ Sebastião Miranda♠ Zita Marinho♡♠ André F. T. Martins† Shay B. Cohen♢

♠ Priberam Labs, Alameda D. Afonso Henriques, 41, 2º, 1000-123 Lisboa, Portugal
♢ School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK
♡ Instituto de Sistemas e Robótica, Instituto Superior Técnico, 1049-001 Lisboa, Portugal
† Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal
♣ Unbabel Lda, Rua Visconde de Santarém, 67-B, 1000-286 Lisboa, Portugal

amm@priberam.com, shashi.narayan@gmail.com, ssm@priberam.com, zam@priberam.com, andre.martins@unbabel.com, scohen@inf.ed.ac.uk

Abstract

We present a new neural model for text summarization that first extracts sentences from a document and then compresses them. The proposed model offers a balance that sidesteps the difficulties in abstractive methods while generating more concise summaries than extractive methods. In addition, our model dynamically determines the length of the output summary based on the gold summaries it observes during training, and does not require length constraints typical to extractive summarization. The model achieves state-of-the-art results on the CNN/DailyMail and Newsroom datasets, improving over current extractive and abstractive methods. Human evaluations demonstrate that our model generates concise and informative summaries. We also make available a new dataset of oracle compressive summaries derived automatically from the CNN/DailyMail reference summaries.

1 Introduction

Text summarization is an important NLP problem with a wide range of applications in data-driven industries (e.g., news, health, and defense). Single document summarization—the task of generating a short summary of a document preserving its informative content (Spärck Jones, 2007)—has been a highly studied research topic in recent years (Nallapati et al., 2016b; See et al., 2017; Fan et al., 2018; Pasunuru and Bansal, 2018).

Modern approaches to single document summarization using neural network architectures have primarily focused on two strategies: extractive and abstractive. The former select a subset of the sentences to assemble a summary (Cheng and Lapata, 2016; Nallapati et al., 2017; Narayan et al., 2018a,c). The latter generates sentences that do not appear in the original document (See et al., 2017; Narayan et al., 2018b; Paulus et al., 2018). Both methods suffer from significant drawbacks: extractive systems are wasteful since they cannot trim the original sentences to fit into the summary, and they lack a mechanism to ensure overall coherence. In contrast, abstractive systems require natural language generation and semantic representation, problems that are inherently harder to solve than just extracting sentences from the original document.

In this paper, we present a novel architecture that attempts to mitigate the problems above via a middle ground, compressive summarization (Martins and Smith, 2009). Our model selects a set of sentences from the input document, and

![Figure 1: Summaries produced by our model. For illustration, the compressive summary shows the removed spans strike-through.](https://github.com/Priberam/exconsumm)  

*Now at Google London.
compresses them by removing unnecessary words, while keeping the summaries informative, concise and grammatical. We achieve this by dynamically modeling the generated summary using a Long Short Term Memory (LSTM; Hochreiter and Schmidhuber, 1997) to produce **summary state representations**. This state provides crucial information to iteratively increment summaries based on previously extracted information. It also facilitates the generation of variable length summaries as opposed to fixed lengths, in previous extractive systems (Cheng and Lapata, 2016; Nallapati et al., 2017; Narayan et al., 2018c; Zhang et al., 2018). Our model can be trained in both extractive (labeling sentences for extraction) or compressive (labeling words for extraction) settings. Figure 1 shows a summary example generated by our model.

Our contributions in this paper are three-fold:

- we present the first end-to-end neural architecture for EXtractive and COmpressive Neu‐ral SUMMarization (dubbed EXCONSUM, see §3),
- we validate this architecture on the CNN/DailyMail and the Newsroom datasets (Hermann et al., 2015; Grusky et al., 2018), showing that our model generates variable-length summaries which correlate well with gold summaries in length and are concise and informative (see §5), and
- we provide a new CNN/DailyMail dataset annotated with automatic compressions for each sentence, and a set of compressed oracle summaries (see §4).

Experimental results show that when evaluated automatically, both the extractive and compressive variants of our model provide state-of-the-art results. Human evaluation further shows that our model is better than previous state-of-the-art systems at generating informative and concise summaries.

### 2 Related Work

Recent work on neural summarization has mainly focused on sequence-to-sequence (seq2seq) architectures (Sutskever et al., 2014), a formulation particularly suited and initially employed for abstractive summarization (Rush et al., 2015). However, state-of-the-art results have been achieved by RNN-based methods which are extractive. They select sentences based on an LSTM classifier that predicts a binary label for each sentence (Cheng and Lapata, 2016), based on ranking using reinforcement learning (Narayan et al., 2018c), or even by training an extractive latent model (Zhang et al., 2018). Other methods rely on an abstractive approach with strongly conditioned generation on the source document (See et al., 2017). In fact, the best results for abstractive summarization have been achieved with models that are more extractive in nature than abstractive, since most of the words in the summary are copied from the document (Gehrmann et al., 2018).

Due to the lack of training corpora, there is almost no work on neural architectures for compressive summarization. Most compressive summarization work has been applied to smaller datasets (Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011; Almeida and Martins, 2013). Other non-neural summarization systems apply this idea to select and compress the summary. Dorr et al. (2003) introduced a method to first extract the first sentence of a news article and then use linguistically-motivated heuristics to iteratively trim parts of it. Durrett et al. (2016) also learns a system that selects textual units to include in the summary and compresses them by deleting word spans guided by anaphoric constraints to improve coherence. Recently, Zhang et al. (2018) trained an abstractive sentence compression model using attention-based sequence-to-sequence architecture (Rush et al., 2015) to map a sentence in the document selected by the extractive model to a sentence in the summary. However, as the sentences in the document and in the summary are not aligned for compression, their compression model is significantly inferior to the extractive model.

In this paper, we propose a novel seq2seq architecture for compressive summarization and demonstrate that it avoids the over-extraction of existing extractive approaches (Cheng and Lapata, 2016; Dlikman and Last, 2016; Nallapati et al., 2016a).

Our model builds on recent approaches to neural extractive summarization as a sequence labeling problem, where sentences in the document are labeled to specify whether or not they should be included in the summary (Cheng and Lapata, 2016; Narayan et al., 2018a). These models often condition their labeling decisions on the document representation only. Nallapati et al. (2017) tries to model the summary as the average representation
of the positively labeled sentences. However, as we show later, this strategy is not the most adequate to ensure summary coherence, as it does not take the order of the selected sentences into account. Our approach addresses this problem by maintaining an LSTM cell to dynamically model the generated summary. To the best of our knowledge, our work is the first to use a model that keeps a state of already generated summary to effectively model variable-length summaries in an extractive setting, and the first to learn a compressive summarizer with an end-to-end approach.

3 Summarization with Summary State Representation

Our model extracts sentences from a given document and further compresses these sentences by deleting words. More formally, we denote a document $D = (s_1, \ldots, s_M)$ as a sequence of $M$ sentences, and a sentence $s_i = (w_{i1}, \ldots, w_{iN})$ as a sequence of $N$ words. We denote by $e(w_{ij})$, $e(s_i)$ and $e(D)$ the embedding of words, sentences and document in a continuous space. We model document summarization as a sequence labeling problem where the labeler transitions between internal states. Each state is dynamically computed based on the context, and it combines an extractive summarizer followed by a compressive one. First, we encode a document in a multi-level approach, to extract the embeddings of words and sentences (“Document Encoder”). Second, we decode these embeddings using a hierarchical “Decision Decoder.” The extractive summarizer labels each sentence $s_i$ with a label $z_i \in \{0, 1\}$ where 1 indicates that the sentence should be included in the final summary and 0 otherwise. An extractive summary is then assembled by selecting all sentences with the label 1. Analogously, the compressive summarizer labels each word $w_{ij}$ with a label $y_{ij} \in \{0, 1\}$, denoting whether the word $j$ in sentence $i$ is included in the summary or not. The final summary is then assembled as the sequence of words $w_{ij}$ for each $z_i = 1$ and $y_{ij} = 1$. See Figures 2 and 3 for an overview of our model. We next describe each of its components in more detail.

3.1 Document Encoder

The document encoder is a two layer biLSTM, one layer encoding each sentence, and the second layer encoding the document. The first layer takes as input the word embeddings $e(w_{ij})$ for each word $j$ in sentence $s_i$, and outputs the hidden representa-
Given that our model operates both at the sentence-level and at the word-level, the decision decoder maintains two state LSTMs denoted by SentStates and WordStates as in Figure 3. For the sentence-level decoder sentences are selected and the state of the summary gets updated by SentStates. For the word-level, all compressed word representations in a sentence are pushed to the word-level layer. In the compressive decoder, words that get selected are pushed onto the WordStates, and once the decoder has reached the end of the sentence, it pushes the output representation of the last state onto the sentence-level layer for the next sentence.

**Extractive Decoder** The extractive decoder selects the sentences that should go to the summary. For each sentence $s_i$ at time step $i$, the decoder takes a decision based on the encoder representation $d_i^e$ and the state of the summary $o_i^s$, computed as follows:

$$o_i^s = \text{SentStates}(\{e(c_k)\}_{k < i, z_k = 1})$$

where the $o_i^s$ is modeled by an LSTM taking as input the already selected and compressed sentences comprising the summary so far $\{e(c_k)\}_{k < i, z_k = 1}$. This way, at each point in time, we have a representation of the summary generated so far, based on the past sentences already processed by the compressive decoder $e(c_{i-1})$ (in WordStates).\(^2\) The summary representation at step $i$ ($o_i^s$) is then used to determine whether to keep or not the current sentence in the summary $(z_i = 1, 0$ respectively). The summarizer state subsumes information about the document, sentence and summary as:

$$p_i = \text{tanh}(W_E[d_i^e; o_i^s] + b^s),$$

where $W_E$ is a model parameter, $o_i^s$ is the dynamic LSTM state, and $b^s$ is a bias term.

This modeling decision is crucial in order to generate variable length summaries. It captures information about the sentences or words already present in the summary, helping in better understanding the “true” length of the summary given the document.

Finally, the summarizer state $p_i$ is used to compute the probability of the action at time $i$ as:

$$p(z_i | p_i) = \frac{\exp\left(W_z p_i + x_{z_i}\right)}{\sum_{z' \in \{0,1\}} \exp\left(W_z p_i + x_{z'}\right)},$$

\(^2\)When using only the extractive model the summary state $o_i^s$ is generated from an LSTM whose inputs correspond to the sentence encoded embeddings $\{e(s_k)\}_{k < i, z_k = 1}$ instead of the previously generated compressed representations $\{e(c_k)\}_{k < i, z_k = 1}$. 

---

**Figure 3:** Decision decoder architecture. Decoder contains an extractive level for sentences (orange) and a compressive level for words (green), using an LSTM to model the summary state. Red diamond shapes represent decision variables $z_i = 1$ if $p(z_i | p_i) > 0.5$ for selecting the sentence $s_i$, and $z_i = 0$ if $p(z_i | p_i) \leq 0.5$ for skipping this sentence. The same for $y_{ij}$ and $p(y_{ij} | q_{ij}) > 0.5$ for deciding over words $w_{ij}$ to keep in the summary.
where \( W_z \) is a model parameter and \( x_z \) is a bias term for the summarizer action \( z \).

We minimize the negative log-likelihood of the observed labels at training time (Dimitroff et al., 2013), where \( \lambda_0^s \) and \( \lambda_s^s \) represent the distribution of each class for the given sentences: \(^3\)

\[
L(\theta^s) = - \sum_{c \in \{0,1\}} \frac{\lambda_c^s}{\sum_{i=1}^{M} \mathbb{1}_{z_i=c} i, z_i=0} \log p(z_i | p_i),
\]

where \( \mathbb{1}_{z_i=c} \) is the indicator function of class \( c \) and \( \theta^s \) represents all the training parameters of the sentence encoder/decoder. At test time, the model emits probability \( p(z_i | p_i) \), which is used as the soft prediction sequentially extracting the sentence \( i \). We admit sentences when \( p(z_i = 1 | p_i) > 0.5 \).

**Compressive Decoder** Our compressive decoder shares its architecture with the extractive decoder. The compressive layer is triggered every time a sentence is selected in the summary and is responsible for selecting the words within each selected sentence. In practice, WordStates LSTM (see Figure 3) is applied hierarchically after the sentence-level decoder, using as input the collected word embeddings so far:

\[
o_{ij}^w = \text{WordStates}(\{e(w_{ik})\}_{k \in \{y_i=1\}}).
\]

After making the selection decision for all words pertaining to a sentence, the final state of the WordStates, \( e(c_i) = o_{iN}^w \), is fed back to SentStates of the extractive level decoder for the consecutive sentence, as depicted in Figure 3.

The word-level summarizer state representation depends on the encoding of words, document and sentence \( d_{ij}^w \), on the dynamic LSTM encoding for the summary based on the selected words (WordStates) \( o_{ij}^w \) and sentences (SentStates) \( o_{ij}^s \):

\[
q_{ij} = \tanh(W_C[d_{ij}^w; o_{ij}^s; o_{ij}^w] + b_w),
\]

where \( W_C \) is a model parameter and \( b_w \) is a bias term. Each action at time step \( j \) is computed by

\[
p(y_{ij} | q_{ij}) = \frac{\exp(W_{y_{ij}} q_{ij} + x_{y_{ij}})}{\sum_{y' \in \{0,1\}} \exp(W_{y'} q_{ij} + x_{y'})},
\]

with parameter \( W_{y_{ij}} \) and bias \( x_{y_{ij}} \). The final loss for the compressive layer is

\[
L(\theta^w) = \sum_{i=1}^{M} z_i \phi(i | \theta^w),
\]

where \( \theta^w \) represents the set of all training parameters of the word-level encoder/decoder, \( \phi(i) \) is the compressive layer loss over \( N \) words:

\[
\phi(i | \theta^w) = - \sum_{c \in \{0,1\}} \frac{\lambda_c^w}{\sum_{i=1}^{M} \mathbb{1}_{y_{ij}=c} i, z_i=0} \log p(y_{ij} | q_{ij}).
\]

The total final loss is then given by the sum of the extractive and compressive counterparts, \( L(\theta) = L(\theta^s) + L(\theta^w) \).

### 4 Experimental Setup

We mainly used the CNN/DailyMail corpus (Hermann et al., 2015) to evaluate our models. We used the standard splits of Hermann et al. (2015) for training, validation, and testing (90,266/1,220/1,093 documents for CNN and 196,961/12,148/10,397 for DailyMail). To evaluate the flexibility of our model, we also evaluated our models on the Newsroom dataset (Grusky et al., 2018), which includes articles from a diverse collection of sources (38 publishers) with different summary style subsets: extractive (Ext.), mixed (Mixed) and abstractive (Abs.). We used the standard splits of Grusky et al. (2018) for training, validation, and testing (331,778/36,322/36,122 documents for Ext., 328,634/35,879/36,006 for Mixed and 332,554/36,380/36,522 for Abs.).

We did not anonymize entities or lower case tokens.

### 4.1 Estimating Oracles

Datasets for training extractive summarization systems do not naturally contain sentence/word-level labels. Instead, they are typically accompanied by abstractive summaries from which extraction labels are extrapolated. We create extractive and compressive summaries prior to training using two types of oracles.

We used an **extractive oracle** to identify the set of sentences which collectively gives the highest ROUGE (Lin and Hovy, 2003) with respect to the gold summary (Narayan et al., 2018c).

To build a **compressive oracle**, we trained a supervised sentence labeling classifier, adapted from
the Transition-Based Chunking Model (Lample et al., 2016), to annotate spans in every sentence that can be dropped in the final summary. We used the publicly released set of 10,000 sentence-compression pairs from the Google sentence compression dataset (Filippova and Altun, 2013; Filippova et al., 2015) for training. After tagging all sentences in the CNN and DailyMail corpora using this compression model, we generated oracle compressive summaries based on the best average of ROUGE-1 (R1) and ROUGE-2 (R2) F1 scores from the combination of all possible sentences and all removals of the marked compression chunks.

To verify the adequacy of our proposed oracles, we show in Table 1 a comparison of their scores. Our compressive oracle achieves much better scores than the extractive oracle, because of its capability to make summaries concise. Moreover, the linguistic quality of these oracles was preserved due to the tagging of the entire span by the sentence compressor trained on the sentence compression dataset.4 We believe that our dataset with oracle compression labels will be of significant interest to the sentence compression and summarization community.

<table>
<thead>
<tr>
<th>Oracle</th>
<th>R1</th>
<th>R2</th>
<th>RL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extractive Oracle</td>
<td>34.67</td>
<td>30.37</td>
<td>50.38</td>
</tr>
<tr>
<td>Compressive Oracle</td>
<td>57.12</td>
<td>32.59</td>
<td>53.27</td>
</tr>
</tbody>
</table>

Table 1: Oracle scores obtained for the CNN and DailyMail testsets. We report ROUGE-1 (R1), ROUGE-2 (R2) and ROUGE-L (RL) F1 scores.

4We show examples of both oracles in Appendix §A.1.

4.2 Training Parameters
The parameters for the loss at the sentence-level were $\lambda_0^s=2$ and $\lambda_1^s=1$ and at the word-level, $\lambda_0^w=1$ and $\lambda_1^w=0.5$. We used LSTMs with $d = 512$ for all hidden layers. We performed mini-batch negative log-likelihood training with a batch size of 2 documents for 5 training epochs. We observed the convergence of the model between the 2nd and the 3rd epochs. It took around 12 hrs on a single GTX 1080 GPU to train. We evaluated our model on the validation set after every 5,000 batches. We trained with Adam (Kingma and Ba, 2015) with an initial learning rate of 0.001. Our system was implemented using DyNet (Neubig et al., 2017).

4.3 Model Evaluation
We evaluated summarization quality using $F_1$ ROUGE (Lin and Hovy, 2003). We report results in terms of unigram and bigram overlap (R1) and (R2) as a means of assessing informativeness, and the longest common subsequence (RL) as a means of assessing fluency.5 In addition to ROUGE, which can be misleading when used as the only means to assess summaries (Schluter, 2017), we also conducted a question-answering based human evaluation to assess the informativeness of our summaries in their ability to preserve key information from the document (Narayan et al., 2018c).6 First, questions are written using the gold summary, we then examined how many questions participants were able to answer by reading system summaries alone, without access to the article.7 Figure 5 shows a set of candidate summaries along with questions used for this evaluation.

4.4 Model and Baselines
We evaluated our model EXCONSUMM in two settings: Extractive (selects sentences to assemble the summary) and Compressive (selects sentences and compresses them by removing unnecessary spans of words). We compared our models against a baseline (LEAD) that selects the first $m$ leading sentences from each document,8 three neural extractive models, and various abstractive models. For the extractive models, we used SUMMARUNNER (Nallapati et al., 2017), since it shares some similarity to our model, REFRESH (Narayan et al., 2018c) trained with reinforcement learning and LATENT (Zhang et al., 2018) a neural architecture that makes use of latent variable to avoid creating oracle summaries. We further compare against LATENT+COMPRESS (Zhang et al., 2018), an extension of the LATENT model that learns to map extracted sentences to final summaries using an attention-based seq2seq model (Rush et al., 2015). All models, unlike ours, extract a fixed number of sentences to assemble their summaries. For abstractive models, we compare against the state-of-the-art models of POINTER+COVERAGE (See et al., 2017), ML+RL (Paulus et al., 2018), and Tan et al. (2017) among others.

5We used pyrouge to compute the ROUGE scores. The parameters we used were “-a -c 95 -m -n 4 -w 1.2”

6We used the CNN/DailyMail QA test set of Narayan et al. (2018c) for evaluation. It includes 20 documents with a total of 71 manually written question-answer pairs.

7See Appendix §A.2 for more details.

8We follow Narayan et al. (2018c) and set $m = 3$ for CNN and 4 for DailyMail. We follow Grusky et al. (2018) and set $m = 2$ for Newsroom.
Comparison with Extractive Systems. ExConSUMM Compressive performs best on the CNN dataset and ExConSUMM Extractive on the DailyMail dataset, probably due to the fact that the CNN dataset is less biased towards extractive methods than the DailyMail dataset (Narayan et al., 2018b). We report similar results on the Newsroom dataset. ExConSUMM Compressive tends to perform better for mixed (Mixed) and abstractive (Abs.) subsets, while ExConSUMM Extractive performs better for the extractive (Ext.) subset. Our experiments demonstrate that our compressive model tends to perform better on the dataset which promotes abstractive summaries.

We find that ExConSUMM Extractive consistently performs better on all metrics when compared to any of the other extractive models, except for the single case where it is narrowly behind LATTENT on R2 (18.6 vs 18.8) for the CNN/DailyMail combined test set. It even outperforms REFRESH, which is trained with reinforcement learning. We hypothesize that its superior performance stems from the ability to generate variable length summaries. REFRESH or LATTENT, on the other hand, always produces a fixed length summary.

Comparison with Compressive System. ExConSUMM Compressive reports superior performance compared to LATTENT+COMPRESS (+4.2 for R1, +2.6 for R2 and +3.1 for RL). Our results demonstrate that our compressive system is more suitable for document summarization. It first selects sentences and then compresses them by removing irrelevant spans of words. It makes use of an advance oracle sentence compressor trained on a dedicated sentence compression dataset (Sec. 4.1). In contrast, LATTENT+COMPRESS naively trains a sequence-to-sequence compressor to map a sentence in the document to a sentence in the summary.

Comparison with Abstractive Systems. Both ExConSUMM Extractive and Compressive outperform most of the abstractive systems including Pointer+Coverage (See et al., 2017). When comparing with more recent methods (Pasunuru and Bansal, 2018; Gehrmann et al., 2018), our model has comparable performance.

Summary Versatility. We evaluate the ability of our model to generate variable length summaries. Table 4 show the Pearson correlation coefficient between the lengths of the human generated summaries against each unbounded model. Our compressive approach obtains the best results, with a Pearson correlation coefficient of 0.72
Table 4: QA evaluations: limited length (Bounded) and full length (Unbounded) summaries. We also show ROUGE scores for the summaries being evaluated. We report the Pearson correlation coefficient between the human and predicted summary lengths.

<table>
<thead>
<tr>
<th>Models</th>
<th>Bounded</th>
<th>Unbounded</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Human QA score</td>
<td>ROUGE</td>
</tr>
<tr>
<td></td>
<td>rank</td>
<td>R1</td>
</tr>
<tr>
<td>LEAD</td>
<td>25.50</td>
<td>4th</td>
</tr>
<tr>
<td>REFRESH</td>
<td>20.88</td>
<td>6th</td>
</tr>
<tr>
<td>LATENT</td>
<td>38.45</td>
<td>2nd</td>
</tr>
<tr>
<td>EXCONSUMM Extractive</td>
<td>36.34</td>
<td>3rd</td>
</tr>
<tr>
<td>EXCONSUMM Compressive</td>
<td>39.44</td>
<td>1st</td>
</tr>
<tr>
<td>Pointer+Coverage</td>
<td>24.51</td>
<td>5th</td>
</tr>
</tbody>
</table>

Figure 4: Word distribution in comparison with the human summaries for CNN dataset. Density curves show the length distributions of human authored and system produced summaries.

(p < 0.001).

Figure 4 also shows the distribution of words per summary for the models where predictions were available. Interestingly, both EXCONSUMM Extractive and Compressive follow the human distribution much better than other extractive systems (LEAD, REFRESH and LATENT), since they are able to generate variable-length summaries depending on the input text. Our compressive model generates a word distribution much closer to the abstractive Pointer+Coverage model but achieves better compression ratio; the summaries generated by Pointer+Coverage contain 59.8 words, while those generated by EXCONSUMM Compressive have 54.3 words on average.

5.2 QA Evaluation

Table 4 shows results from our question answering based human evaluation. We elicited human judgements in two settings: the “Unbounded”, where participants were shown the full system produced summaries; and the “Bounded”, where participants were shown summaries that were limited to the same size as the gold summaries.

For the “Unbounded” setting, the output summaries produced by REFRESH were able to answer most of the questions correctly, our Compressive and Extractive systems were placed at the 2nd and 3rd places respectively.

We observed that our systems were able to produce more concise summaries than those produced by REFRESH (avg. length in words: 76.0 for REFRESH, 56.2 for EXCONSUMM Extractive and 54.3 for EXCONSUMM Compressive; see Figure 4). REFRESH is prone to generating verbose summaries, consequently it has an advantage of accumulating more information. In the “Bounded” setting, we aim to reduce this unfair advantage. Scores are overall lower since the summary sizes are truncated to gold size. The EXCONSUMM Compressive summaries rank first and can answer 39.44% of questions correctly. EXCONSUMM Extractive retains its 3rd place answering 36.34% of questions correctly. These results demonstrate that our models generate concise and informative summaries that correlate well with the human summary lengths.

---

9We carried out pairwise comparisons between all models to assess whether system differences are statistically significant. We found that there is no statistically significant difference between REFRESH and EXCONSUMM Compressive. We use a one-way ANOVA with posthoc Tukey HSD tests with p < 0.01. The differences among LATENT and both variants of EXCONSUMM, and between LEAD and Pointer+Coverage are also statistically insignificant. All other differences are statistically significant.

10The differences among both variants of EXCONSUMM and LATENT, and among LEAD, REFRESH and Pointer+Coverage are statistically insignificant. All other differences are statistically significant. We use a one-way ANOVA with posthoc Tukey HSD tests with p < 0.01.

11App. §A.2 shows more examples of our summaries.
**LEAD**
- (CNN) A top al Qaeda in the Arabian Peninsula leader—who a few years ago was in a U.S. detention facility—was among five killed in an airstrike in Yemen, the terror group said, showing the organization is vulnerable even as Yemen appears close to civil war.
- Ibrahim al-Rubaish died Monday night in what AQAP’s media wing, Al-Malahem Media, called a “crusader airstrike.”
- The Al-Malahem Media obituary characterized al-Rubaish as a religious scholar and combat commander.

**REFRESH**
- (CNN) A top al Qaeda in the Arabian Peninsula leader—who a few years ago was in a U.S. detention facility—was among five killed in an airstrike in Yemen, the terror group said, showing the organization is vulnerable even as Yemen appears close to civil war.
- Ibrahim al-Rubaish died Monday night in what AQAP’s media wing, Al-Malahem Media, called a “crusader airstrike.”
- Al-Rubaish was once held by the U.S. government at its detention facility in Guantanamo Bay, Cuba.

**LATENT**
- (CNN) A top al Qaeda in the Arabian Peninsula leader—who a few years ago was in a U.S. detention facility—was among five killed in an airstrike in Yemen, the terror group said, showing the organization is vulnerable even as Yemen appears close to civil war.
- Ibrahim al-Rubaish died Monday night in what AQAP’s media wing, Al-Malahem Media, called a “crusader airstrike.”
- A Yemeni Defense Ministry official and two Yemeni national security officials not authorized to speak on record confirmed that al-Rubaish had been killed, but could not specify how he died.

**EXCONSUMM Extractive**
- (CNN) A top al Qaeda in the Arabian Peninsula leader—who a few years ago was in a U.S. detention facility—was among five killed in an airstrike in Yemen, the terror group said, showing the organization is vulnerable even as Yemen appears close to civil war.
- Ibrahim al-Rubaish died Monday night in what AQAP’s media wing, Al-Malahem Media, called a “crusader airstrike.”

**EXCONSUMM Compressive**
- A top al Qaeda in the Arabian Peninsula leader—who a few years ago was in a U.S. detention facility—was among five killed in an airstrike in Yemen, the terror group said, showing the organization is vulnerable even as Yemen appears close to civil war.
- Ibrahim al-Rubaish died Monday night in what AQAP’s media wing, Al-Malahem Media, called a “crusader airstrike.”

**Pointer+Coverage**
- Ibrahim al-Rubaish was among a number of detainees who sued the administration of then-president George W. Bush to challenge the legality of their confinement in Gitmo. • al-Rubaish was once held by the U.S. government at its detention facility in Guantanamo Bay, Cuba.

**GOLD**
- AQAP says a “crusader airstrike” killed Ibrahim al-Rubaish.
- Al-Rubaish was once detained by the United States in Guantanamo Bay, Cuba.

**Question-Answer Pairs**
- Who said that an airstrike killed Ibrahim al-Rubaish? (AQAP) • What was the airstrike called? (crusader airstrike) • Where was Ibrahim al-Rubaish once detained? (Guantanamo)

Figure 5: Example output summaries on the CNN/DailyMail dataset, gold standard summary, and corresponding questions. The questions are manually written using the GOLD summary. The same EXCONSUMM summaries are shown in Figure 1, but the strike-through spans are now removed.

### 5.3 Summary State Representation

Next, we performed an ablation study to investigate the importance of the summary state representation $o^*_i$ w.r.t. the quality of the overall summary. We tested against a **STATE AVERAGING** variant, where we replace $o^*_i$ by a weighted average, analogous to Nallapati et al. (2017), $o^{avg}_i = \sum_{i=1}^{t-1} e(s_i) p(z_i | p^{i}_{avg})$, where $p^{avg}_i$ has the same form as $p_i$ but depends recursively on the previous summary state $o^{avg}_{i-1}$. Table 5 shows that using an LSTM state $o^*_i$ to model the current sentences in the summary is very important. The other ablation study shows how learning to extract and compress in a disjoint approach (EXCONSUMM Ext+Comp oracle) performs against a joint learning approach (EXCONSUMM Compressive). We compared summaries generated from our best extractive model and compressed them with a compressive oracle. Our joint learning model achieves the best performance in all metrics compared with the other ablations, suggesting that joint learning and using a summary state representation is beneficial for summarization.

### 6 Conclusions

We developed EXCONSUMM, a novel summarization model to generate variable length extractive and compressive summaries. Experimental results show that the ability of our model to learn a dynamic representation of the summary produces summaries that are informative, concise, and correlate well with human generated summary lengths. Our model outperforms state-of-the-art extractive and most of abstractive systems on the CNN and DailyMail datasets, when evaluated automatically, and through human evaluation for the bounded scenario. We further obtain state-of-the-art results on Newsroom, a more abstractive summary dataset.

### Acknowledgments

This work is supported by the EU H2020 SUMMA project (grant agreement N° 688139), by Lisbon Regional Operational Programme (Lisboa 2020), under the Portugal 2020 Partnership Agreement, through the European Regional Development Fund (ERDF), within project INSIGHT (N° 033869), by the European Re-
search Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contracts UID/EEA/50008/2019 and CMUPERI/TIC/0046/2014 (GoLocal).

References


Alexander Dlikman and Mark Last. 2016. Using machine learning methods and linguistic features in single-document extractive summarization. In DMNLP@PKDD/ECML.


A Appendices

A.1 Estimating Summary Oracles

We describe our method to estimate extractive and compressive oracle summaries prior to training using two types of oracles. We build these oracles in order to train our model with a supervised objective by minimizing a negative log-likelihood function. We create documents annotated with sentence-level and word-level extraction labels, which correspond to the gold values of both variables $z_i$ and $y_{ij}$, respectively.

**Extractive Oracle.** We followed Narayan et al. (2018c) and identified the set of sentences which collectively give the highest ROUGE (Lin and Hovy, 2003) with respect to the gold summary. More concretely, we assembled candidate summaries efficiently by first selecting $p$ sentences from the document which on their own have high ROUGE scores. We then generated all possible combinations of $p$ sentences subject to maximum length $m$ (3 for CNN and 4 for DailyMail) and evaluated them against the gold summary. We select the summary with the highest mean of ROUGE-1, ROUGE-2, and ROUGE-L F1 scores.

**Compressive Oracle.** The primary challenge in building a compressive oracle lies in preserving the grammaticality of compressed sentences. Following the sentence compression literature (McDonald, 2006; Clarke and Lapata, 2008; Berg-Kirkpatrick et al., 2011; Filippova and Altun, 2013; Filippova et al., 2015), we train a supervised neural model to annotate spans in every sentence that can be dropped. In particular, we trained a supervised sentence labeling classifier adapted from Lample et al. (2016). To train our classifier, we used the publicly released set of 10,000 sentence-compression pairs from the Google sentence compression dataset (Filippova et al., 2015; Filippova and Altun, 2013). We removed the first 1,000 sentences as the development set and used the remaining ones as the training set.

After training our classifier for 30 epochs, it achieved a per-sentence accuracy of 21%, a word-based F-1 score of 78% and a compression ratio of 0.38. The parameters for the model were: 2 layers, dropout of 0.1, hidden dimension of size 400, action dimension of 20 and relation dimension of 20. We used the One Billion Word Benchmark corpus (Chelba et al., 2013) to train word embeddings with the skip-gram model (Mikolov et al., 2013) using context window size 6, negative sampling size 10, and hierarchical softmax 1. Same embeddings were used to train our summarization model also. For details of the evaluation metrics, please see Filippova et al. (2015).

After tagging all sentences in the CNN and DailyMail corpora using this compression model, we generated oracle compressive summaries based on the best average of ROUGE-1 and ROUGE-2 F1 scores from the combination of all possible sen-

---

Figure 6: Examples of our estimated oracle summaries along with the reference summary for the CNN and DailyMail datasets. For illustration, the compressive oracle shows the removed spans strike-through.
Kanye West has settled a lawsuit with a paparazzi after the hip-hop star attacked him and tried to wrestle his camera from him in July 2013 at Los Angeles International Airport. The photographer, Daniel Ramos, had filed the civil suit against West after the hip-hop star attacked him and tried to wrestle his camera from him in July 2013 at Los Angeles International Airport.

EXCONSUMM BOW (Baseline)

• West pleaded no contest last year to a misdemeanor count of battery after the hip-hop star attacked him and tried to wrestle his camera from him in July 2013 at Los Angeles International Airport.

EXCONSUMM Compressive

• West pleaded no contest last year to a misdemeanor count of battery after the hip-hop star attacked him and tried to wrestle his camera from him in July 2013 at Los Angeles International Airport.

Additionally, we also experimented with a bag-of-words oracle (BOW oracle), that serves as a baseline for creating a compressive oracle, in which labels are generated by simply dropping words if they do not appear in the gold summary. Unsurprisingly, oracle sentences compressed with this method are often ungrammatical (see Figure 6).

Our model EXCONSUMM trained with the BOW oracle (EXCONSUMM BOW) often score higher than the scores of the compressive model as shown in Table 6. However, looking at the example summaries in Figure 7, we find that the BOW compressive model is incapable of generating a fluent or grammatical summary. The EXCONSUMM Compressive summary, on the other hand, is fluent and grammatical. Our summary LSTMs (WordStates and SentStates) can preserve the fluency of the summaries if trained with the fluent Compressive oracle. This is not guaranteed when using the BOW oracle.

### A.2 Human Evaluation Experiment Design

The main assumption behind this evaluation is that the gold summary highlights the most important content of the document. Based on this assumption, the questions are written using the GOLD content of the document. Based on this assumption, the questions are written using the gold summary as a guide to summarizing the document. For this study, we used the same 20 documents (10 from CNN and 10 from DailyMail tests) with an accompanying set of questions based on the gold summary from Narayan et al. (2018c).\(^\text{12}\) We examined how many questions participants were able to answer by reading system summaries alone, without access to the article. The more questions a system can help to answer, the better it is at summarizing the document as a whole. We collected answers from five different participants for each summary and system pair. We marked a correct answer with a score of one, partially correct answers with a score of 0.5, and zero otherwise, the final score is an average of all these scores. Answers were elicited using Amazon’s Mechanical Turk crowd-sourcing platform. Examples of systems summaries used for this evaluation are shown in Figures 5, 8, 9, 10, 11, 12 and 13.

---

\(^{12}\)The test set for the QA evaluation is publicly available at [https://github.com/EdinburghNLP/Refresh](https://github.com/EdinburghNLP/Refresh).
LEAD
• (CNN) Seven people—including Illinois State University associate men’s basketball coach Torrey Ward and deputy athletic director Aaron Leetch—died when their small plane crashed while heading back from the NCAA tournament final.
• The aircraft went down overnight Monday about 2 miles east of the Central Illinois Regional Airport in Bloomington, McLean County Sheriff’s Office Sgt. Bill Tate said.
• That’s about 5 miles from the campus of Illinois State, where Ward and Leetch both worked.

REFRESH
• (CNN) Seven people—including Illinois State University associate men’s basketball coach Torrey Ward and deputy athletic director Aaron Leetch—died when their small plane crashed while heading back from the NCAA tournament final.
• The aircraft went down overnight Monday about 2 miles east of the Central Illinois Regional Airport in Bloomington, McLean County Sheriff’s Office Sgt. Bill Tate said.
• The plane was coming back from the NCAA Final Four championship game in Indianapolis, according to Illinois State athletics spokesman John Twork.

LATENT
• (CNN) Seven people—including Illinois State University associate men’s basketball coach Torrey Ward and deputy athletic director Aaron Leetch—died when their small plane crashed while heading back from the NCAA tournament final.
• The plane was coming back from the NCAA Final Four championship game in Indianapolis, according to Illinois State athletics spokesman John Twork.
• The aircraft went down overnight Monday about 2 miles east of the Central Illinois Regional Airport in Bloomington, McLean County Sheriff’s Office Sgt. Bill Tate said.

EXCONSUMM Extractive
• (CNN) Seven people—including Illinois State University associate men’s basketball coach Torrey Ward and deputy athletic director Aaron Leetch—died when their small plane crashed while heading back from the NCAA tournament final.
• The plane was coming back from the NCAA Final Four championship game in Indianapolis, according to Illinois State athletics spokesman John Twork.

EXCONSUMM Compressive
• Seven people died their small plane crashed while heading back from the NCAA tournament final.
• The aircraft went down overnight Monday about 2 miles east of the Central Illinois Regional Airport in Bloomington.
• The plane was coming back from the NCAA Final Four championship game.

Pointer+Coverage
• Illinois State University associate men’s basketball coach Torrey Ward and deputy athletic director Aaron Leetch died when their small plane crashed while heading back from the NCAA tournament final.
• The aircraft went down overnight Monday about 2 miles east of the Central Illinois Regional Airport in Bloomington, Illinois.
• It was not immediately known who else was on the aircraft, which the National Transportation Safety Board tweeted was a Cessna 414.
• There’s also a picture of a small plane with the words, “my ride to the game was n’t bad #indy2015f4”.

GOLD
• The crashed plane was a Cessna 414, National Transportation Safety Board reports
• Coach Torrey Ward, administrator Aaron Leetch among the 7 killed in the crash
• The plane crashed while coming back from the NCAA title game in Indianapolis

Question-Answer Pairs
• What type of plane crashed? (Cessna 414)
• Who are confirmed dead in the crash? (Coach Torrey Ward and administrator Aaron Leetch)
• How many people in total died in the crash? (7 people)
• The plane crashed while coming back from where? (The NCAA title game in Indianapolis)
A hedgehog sniffing around in the dusk was once a common sight - but experts warn it may soon become a thing of the past. One in five people have never seen a hedgehog in their gardens, according to a wildlife survey. And of those who do spot the tiny animals, only a quarter see them frequently, the RSPB found. The startling figures confirm fears that the small British mammal is suffering a huge decline.

One in five people have never seen a hedgehog in their gardens, according to a wildlife survey. And of those who do spot the tiny animals, only a quarter see them frequently, the RSPB found. The startling figures confirm fears that the small British mammal is suffering a huge decline. There are thought to be less than 1 million hedgehogs living in this country today, an estimated 30 per cent drop since 2013.

One in five people have never seen a hedgehog in their gardens, according to a new wildlife survey. The startling figures confirm fears that the small British mammal is suffering a huge decline.

There are thought to be less than 1 million hedgehogs living in this country today, an estimated 30 per cent drop since 2013.

One in five people have never seen a hedgehog in their gardens. Of those who do spot the tiny animals, only a quarter see them frequently. The startling figures confirm fears that the small British mammal is suffering a huge decline.

One in five people have never seen a hedgehog in their back gardens. Only a quarter of those who do admit to seeing the animals frequently. Wildlife survey suggested the small British mammal is in huge decline. There are thought to be less than 1 million hedgehogs in the country.

How many people have never seen a hedgehog in their back gardens? (One in five people) Who conducted this survey? (Wildlife survey) How many hedgehogs are thought to be left in the country? (less than 1 million)

Figure 9: Example output summaries on the CNN/DailyMail dataset, gold standard summary, and corresponding questions.
LEAD
• (CNN) Blinky and Pinky on the Champs Elysees?
• Inky and Clyde running down Broadway?
• Power pellets on the Embarcadero?

REFRESH
• Leave it to Google to make April Fools’ Day into throwback fun by combining Google Maps with Pac-Man.
• The massive tech company is known for its impish April Fools’ Day pranks, and Google Maps has been at the center of a few, including a Pokemon Challenge and a treasure map.
• This year the company was a day early to the party, rolling out the Pac-Man game Tuesday.

LATENT
• Leave it to Google to make April Fools’ Day into throwback fun by combining Google Maps with Pac-Man.
• (CNN) Blinky and Pinky on the Champs Elysees? Inky and Clyde running down Broadway? Power pellets on the Embarcadero?
• Twitterers have been tickled by the possibilities, playing Pac-Man in Manhattan, on the University of Illinois quad, in central London and down crooked Lombard Street in San Francisco, among many locations:

EXCONSUMM Extractive
• Leave it to Google to make April Fools’ Day into throwback fun by combining Google Maps with Pac-Man.
• It’s easy to play: Simply pull up Google Maps on your desktop browser, click on the Pac-Man icon on the lower left, and your map suddenly becomes a Pac-Man course.

EXCONSUMM Compressive
• Leave it to Google to make April Fools’ Day into throwback fun by combining Google Maps with Pac-Man.
• The tech company is known for its April Fools’ Day pranks.

Pointer+Coverage
• The massive tech company is known for its impish April Fools’ Day pranks, and Google Maps has been at the center of a few, including a Pokemon challenge and a treasure map.
• It’s easy to play: simply pull up Google Maps on your desktop browser.

GOLD
• Google Maps has a temporary Pac-Man function
• Google has long been fond of April Fools’ Day pranks and games
• Many people are turning their cities into Pac-Man courses

Question-Answer Pairs
• What function does Google Maps have? (Pac-Man)
• What has Google been long fond of? (April Fools’ Day pranks and games)
• What are many people turning their cities into? (Pac-Man courses)

Figure 10: Example output summaries on the CNN/DailyMail dataset, gold standard summary, and corresponding questions.
<table>
<thead>
<tr>
<th>LEAD</th>
<th>(CNN) Somewhere over the rainbow, people on the Internet are losing their minds.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Is it real?</td>
</tr>
<tr>
<td></td>
<td>After the New York area received a large amount of rain, four rainbows stretched across the early morning sky on Tuesday.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REFRESH</th>
<th>(CNN) Somewhere over the rainbow, people on the Internet are losing their minds.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>After the New York area received a large amount of rain, four rainbows stretched across the early morning sky on Tuesday.</td>
</tr>
<tr>
<td></td>
<td>Amanda Curtis, CEO of a fashion company in New York, snapped the lucky shot.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LATENT</th>
<th>Amanda Curtis, CEO of a fashion company in New York, snapped the lucky shot.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>After the New York area received a large amount of rain, four rainbows stretched across the early morning sky on Tuesday.</td>
</tr>
<tr>
<td></td>
<td>CNN iReporter Yosemitebear Vasquez posted a video to YouTube in 2010 reacting to a double rainbow he spotted in Yosemite National Park. The video has since garnered over 40 million views.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ExConSUMM Extractive</th>
<th>Four rainbows stretched across the early morning sky on Tuesday.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Amanda Curtis, CEO of a fashion company in New York, snapped the lucky shot.</td>
</tr>
<tr>
<td></td>
<td>The video has since garnered over 40 million views.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ExConSUMM Compressive</th>
<th>Four rainbows stretched across the early morning sky on Tuesday.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Amanda Curtis, CEO of a fashion company in New York, snapped the lucky shot.</td>
</tr>
<tr>
<td></td>
<td>The video has since garnered over 40 million views.</td>
</tr>
</tbody>
</table>

| Pointer+Coverage | She posted the picture to Twitter, and within a few hours, it had already received hundreds of retweets. |

<table>
<thead>
<tr>
<th>GOLD</th>
<th>Amanda Curtis, CEO of a fashion company in New York, posted a picture of four rainbows to Twitter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“I had a small moment of awe,” she said</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question-Answer Pairs</th>
<th>Who posted a picture to Twitter? (Amanda Curtis)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>What did the picture show? (four rainbows)</td>
</tr>
<tr>
<td></td>
<td>What is the profession of the person who posted this picture? (CEO of a fashion company in New York)</td>
</tr>
</tbody>
</table>

Figure 11: Example output summaries on the CNN/DailyMail dataset, gold standard summary, and corresponding questions.
• The Fulham fans in the Jimmy Steed Stand applauded their team at the final whistle.
• It was not the victory manager Kit Symons had called for but a point away to Charlton probably secures their future in the Championship next season.
• They are eight points clear of Millwall and bar a miraculous resurgence from one of the bottom three sides will stay up but the fact relegation is still mathematically feasible for a club that were in the Premier League last season is alarming.
• A vertiginous decline, just one victory in their last seven games had seen them dragged back into a relegation battle and after a painful 4-1 trouncing by bitter rivals Brentford last week, Symons was looking for a quick response from his players.

• Ross McCormack gave Fulham the lead after eight minutes at The Valley
• But Johann Gudmundsson leveled the scores less than ten minutes later
• Scott Parker was booed on his return to club, 11 years after he left
• Share of the points in London leaves Charlton in 11th and Fulham in 20th

Figure 12: Example output summaries on the CNN/DailyMail dataset, gold standard summary, and corresponding questions.
LEAD
• (CNN) You probably never knew her name, but you were familiar with her work.
• Betty Whitehead Willis, the designer of the iconic “Welcome to Fabulous Las Vegas” sign, died over the weekend.
• She was 91.

REFRESH
• Betty Whitehead Willis, the designer of the iconic “Welcome to Fabulous Las Vegas” sign, died over the weekend.
• Willis played a major role in creating some of the most memorable neon work in the city.
• Willis visited the Neon Museum in 2013 to celebrate her 90th birthday.

LATENT
• Betty Whitehead Willis, the designer of the iconic “Welcome to Fabulous Las Vegas” sign, died over the weekend. She was 91.
• The Neon Museum also credits her with designing the signs for Moulin Rouge Hotel and Blue Angel Motel.
• Willis visited the Neon Museum in 2013 to celebrate her 90th birthday.

EXCONSUMM Extractive
• Betty Whitehead Willis, the designer of the iconic “Welcome to Fabulous Las Vegas” sign, died over the weekend.
• Willis visited the Neon Museum in 2013 to celebrate her 90th birthday.

EXCONSUMM Compressive
• Betty Whitehead Willis died over the weekend.
• Willis played a major role in creating some of the most memorable neon work in the city.
• Willis visited the Neon Museum in 2013 to celebrate her 90th birthday.

Pointer+Coverage
• Betty Whitehead Willis, the designer of the iconic “Welcome to Fabulous Las Vegas, died over the weekend.
• She was 91.
• Willis never trademarked her most-famous work, calling it “my gift to the city”.

GOLD
• Willis never trademarked her most-famous work, calling it “my gift to the city”
• She created some of the city’s most famous neon work

Question-Answer Pairs
• What was Willis’ most-famous work called? (my gift to the city)
• What did Willis create in the city? (City’s most famous neon work)

Figure 13: Example output summaries on the CNN/DailyMail dataset, gold standard summary, and corresponding questions.