Abstract

Single document summarization is the task of producing a shorter version of a document while preserving its principal information content. In this paper we conceptualize extractive summarization as a sentence ranking task and propose a novel training algorithm which globally optimizes the ROUGE evaluation metric through a reinforcement learning objective. We use our algorithm to train a neural summarization model on the CNN and DailyMail datasets and demonstrate experimentally that it outperforms state-of-the-art extractive and abstractive systems when evaluated automatically and by humans.

1 Introduction

Automatic summarization has enjoyed wide popularity in natural language processing due to its potential for various information access applications. Examples include tools which aid users navigate and digest web content (e.g., news, social media, product reviews), question answering, and personalized recommendation engines. Single document summarization — the task of producing a shorter version of a document while preserving its information content — is perhaps the most basic of summarization tasks that have been identified over the years (see Nenkova and McKeown, 2011 for a comprehensive overview).

Modern approaches to single document summarization are data-driven, taking advantage of the success of neural network architectures and their ability to learn continuous features without recourse to preprocessing tools or linguistic annotations. Abstractive summarization involves various text rewriting operations (e.g., substitution, deletion, reordering) and has been recently framed as a sequence-to-sequence problem (Sutskever et al., 2014). Central in most approaches (Rush et al., 2015; Chen et al., 2016; Nallapati et al., 2016; See et al., 2017; Tan and Wan, 2017; Paulus et al., 2017) is an encoder-decoder architecture modeled by recurrent neural networks. The encoder reads the source sequence into a list of continuous-space representations from which the decoder generates the target sequence. An attention mechanism (Bahdanau et al., 2015) is often used to locate the region of focus during decoding.

Extractive systems create a summary by identifying (and subsequently concatenating) the most important sentences in a document. A few recent approaches (Cheng and Lapata, 2016; Nallapati et al., 2017; Narayan et al., 2017; Yasunaga et al., 2017) conceptualize extractive summarization as a sequence labeling task in which each label specifies whether each document sentence should be included in the summary. Existing models rely on recurrent neural networks to derive a meaning representation of the document which is then used to label each sentence, taking the previously labeled sentences into account. These models are typically trained using cross-entropy loss in order to maximize the likelihood of the ground-truth labels and do not necessarily learn to rank sentences based on their importance due to the absence of a ranking-based objective. Another discrepancy comes from the mismatch between the learning objective and the evaluation criterion, namely ROUGE (Lin and Hovy, 2003), which takes the entire summary into account.

In this paper we argue that cross-entropy training is not optimal for extractive summarization. Models trained this way are prone to generating verbose summaries with unnecessarily long sentences and redundant information. We propose to overcome these difficulties by globally optimizing the ROUGE evaluation metric and learning to rank sentences for summary generation through a reinforcement learning objective. Similar to previous work (Cheng and Lapata, 2016; Narayan et al., 2017; Nallapati et al., 2017), our neural summarization model consists of a hierarchical document encoder and a hierarchical sentence extractor. During training, it combines the maximum-
likelihood cross-entropy loss with rewards from policy gradient reinforcement learning to directly optimize the evaluation metric relevant for the summarization task. We show that this global optimization framework renders extractive models better at discriminating among sentences for the final summary; a sentence is ranked high for selection if it often occurs in high scoring summaries.

We report results on the CNN and DailyMail news highlights datasets (Hermann et al., 2015) which have been recently used as testbeds for the evaluation of neural summarization systems. Experimental results show that when evaluated automatically (in terms of ROUGE), our model outperforms state-of-the-art extractive and abstractive systems. We also conduct two human evaluations in order to assess (a) which type of summary participants prefer (we compare extractive and abstractive systems) and (b) how much key information from the document is preserved in the summary (we ask participants to answer questions pertaining to the content in the document by reading system summaries). Both evaluations overwhelmingly show that human subjects find our summaries more informative and complete.

Our contributions in this work are three-fold: a novel application of reinforcement learning to sentence ranking for extractive summarization; corroborated by analysis and empirical results showing that cross-entropy training is not well-suited to the summarization task; and large scale user studies following two evaluation paradigms which demonstrate that state-of-the-art abstractive systems lag behind extractive ones when the latter are globally trained.

2 Summarization as Sentence Ranking

Given a document $D$ consisting of a sequence of sentences $(s_1, s_2, \ldots, s_n)$, an extractive summarizer aims to produce a summary $S$ by selecting $m$ sentences from $D$ (where $m < n$). For each sentence $s_i \in D$, we predict a label $y_i \in \{0, 1\}$ (where 1 means that $s_i$ should be included in the summary) and assign a score $p(y_i|s_i, D, \theta)$ quantifying $s_i$’s relevance to the summary. The model learns to assign $p(1|s_i, D, \theta) > p(1|s_j, D, \theta)$ when sentence $s_i$ is more relevant than $s_j$. Model parameters are denoted by $\theta$. We estimate $p(y_i|s_i, D, \theta)$ using a neural network model and assemble a summary $S$ by selecting $m$ sentences with top $p(1|s_i, D, \theta)$ scores.

Our architecture resembles those previously proposed in the literature (Cheng and Lapata, 2016; Nallapati et al., 2017; Narayan et al., 2017). The main components include a sentence encoder, a document encoder, and a sentence extractor (see the left block of Figure 1) which we describe in more detail below.

Sentence Encoder A core component of our model is a convolutional sentence encoder which encodes sentences into continuous representations. In recent years, CNNs have proven useful for various NLP tasks (Collobert et al., 2011; Kim, 2014; Kalchbrenner et al., 2014; Zhang et al., 2015; Lei et al., 2015; Kim et al., 2016; Cheng and Lapata, 2016) because of their effectiveness in identifying salient patterns in the input (Xu et al., 2015). In the case of summarization, CNNs can identify named-entities and events that correlate with the gold summary.

We use temporal narrow convolution by applying a kernel filter $K$ of width $h$ to a window of $h$ words in sentence $s$ to produce a new feature. This filter is applied to each possible window of words in $s$ to produce a feature map $f \in R^{k-h+1}$ where $k$ is the sentence length. We then apply max-pooling over time over the feature map $f$ and take the maximum value as the feature corresponding to this particular filter $K$. We use multiple kernels of various sizes and each kernel multiple times to construct the representation of a sentence. In Figure 1, kernels of size 2 (red) and 4 (blue) are applied three times each. Max-pooling over time yields two feature lists $f^{K_2}$ and $f^{K_4} \in R^3$. The final sentence embeddings have six dimensions.

Document Encoder The document encoder composes a sequence of sentences to obtain a document representation. We use a recurrent neural network with Long Short-Term Memory (LSTM) cells to avoid the vanishing gradient problem when training long sequences (Hochreiter and Schmidhuber, 1997). Given a document $D$ consisting of a sequence of sentences $(s_1, s_2, \ldots, s_n)$, we follow common practice and feed sentences in reverse order (Sutskever et al., 2014; Li et al., 2015; Filipova et al., 2015; Narayan et al., 2017). This way we make sure that the network also considers the top sentences of the document which are particularly important for summarization (Rush et al., 2015; Nallapati et al., 2016).

Sentence Extractor Our sentence extractor sequentially labels each sentence in a document with 1 (relevant for the summary) or 0 (otherwise). It is implemented with another RNN with LSTM cells and a softmax layer. At time $t_i$, it reads sentence $s_i$ and makes a binary prediction, conditioned on the document representation (obtained
Cross-entropy training leads to two kinds of discrepancies in the model. The first discrepancy comes from the disconnect between the task definition and the training objective. While MLE in Equation (1) aims to maximize the likelihood of the ground-truth labels, the model is (a) expected to rank sentences to generate a summary and (b) evaluated using ROUGE at test time. The second discrepancy comes from the reliance on ground-truth labels. Document collections for training summarization systems do not naturally contain labels indicating which sentences should be extracted. Instead, they are typically accompanied by abstractive summaries from which sentence-level labels are extrapolated. Cheng and Lapata (2016) follow Woodsend and Lapata (2010) in adopting a rule-based method which assigns labels to each sentence in the document individually based on their semantic correspondence with the gold summary (see the fourth column in Table 1). An alternative method (Svore et al., 2007; Cao et al., 2016; Nallapati et al., 2017) identifies the set of sentences which collectively gives the highest ROUGE with respect to the gold summary. Sentences in this set are labeled with 1 and 0 otherwise (see the column 5 in Table 1).

Labeling sentences individually often generates too many positive labels causing the model to overfit the data. For example, the document in Table 1 has 12 positively labeled sentences out of 31 in total (only first 10 are shown). Collective labels present a better alternative since they only per-
A debilitating, mosquito-borne virus called Chikungunya has made its way to Florida. But a major outbreak in the Caribbean this year – with more than 100,000 cases – has alarmed public health officials. "We haven’t had any locally transmitted cases in the U.S. thus far," Nasci said. The virus, which can cause joint pain and arthritis-like symptoms, has been on North Carolina’s radar for some time. About 25 to 28 infected travelers bring it to the United States each year, said Roger Nasci, chief of the CDC’s Arboviral Disease Branch in the Division of Vector-Borne Diseases. "We haven’t had any locally transmitted cases in the U.S. thus far," Nasci said.

But a major outbreak in the Caribbean this year – with more than 100,000 cases reported – has health officials concerned. Experts say American tourists are bringing Chikungunya back home, and it’s just a matter of time before it starts to spread within the United States. After all, the Caribbean is a popular one with American tourists, and summer is fast approaching. "So far this year we’ve recorded eight travel-associated cases, and seven of them have come from countries in the Caribbean where we know the virus is being transmitted," Nasci said.

Other states have also reported cases of Chikungunya. The Tennessee Department of Health said the state has had multiple cases of the virus in people who have traveled to the Caribbean. The virus is not deadly, but it can be painful, with symptoms lasting for weeks. Those with weak immune systems, such as the elderly, are more likely to suffer from the virus’ side effects than those who are healthier. North Carolina reports first case of mosquito-borne virus called Chikungunya. Chikungunya is primarily found in Africa, East Asia and the Caribbean islands. Virus is not deadly, but it can be painful, with symptoms lasting for weeks.

### Table 1

<table>
<thead>
<tr>
<th>Sent pos</th>
<th>CNN article</th>
<th>Sent-level</th>
<th>Individual Oracle</th>
<th>Collective Oracle</th>
<th>Multiple Collective Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A debilitating, mosquito-borne virus called Chikungunya has made its way to North Carolina, health officials say.</td>
<td>21.2</td>
<td>1</td>
<td>1</td>
<td>(0,11,13): 59.3</td>
</tr>
<tr>
<td>1</td>
<td>It’s the state’s first reported case of the virus.</td>
<td>18.1</td>
<td>1</td>
<td>0</td>
<td>(11,13): 57.2</td>
</tr>
<tr>
<td>2</td>
<td>The patient was likely infected in the Caribbean, according to the Forsyth County Department of Public Health.</td>
<td>11.2</td>
<td>1</td>
<td>0</td>
<td>(0,11,13): 57.1</td>
</tr>
<tr>
<td>3</td>
<td>Chikungunya is primarily found in Africa, East Asia and the Caribbean islands, but the Centers for Disease Control and Prevention has been watching the virus, for fear that it could take hold in the United States – much like West Nile did more than a decade ago.</td>
<td>35.6</td>
<td>1</td>
<td>0</td>
<td>(3,11,13): 55.0</td>
</tr>
<tr>
<td>4</td>
<td>The virus, which can cause joint pain and arthritis-like symptoms, has been on the U.S. public health radar for some time.</td>
<td>16.7</td>
<td>1</td>
<td>0</td>
<td>(1,13): 54.5</td>
</tr>
<tr>
<td>5</td>
<td>About 25 to 28 infected travelers bring it to the United States each year, said Roger Nasci, chief of the CDC’s Arboviral Disease Branch in the Division of Vector-Borne Diseases.</td>
<td>9.7</td>
<td>0</td>
<td>0</td>
<td>(0,13): 51.3</td>
</tr>
<tr>
<td>6</td>
<td>&quot;We haven’t had any locally transmitted cases in the U.S. thus far,&quot; Nasci said.</td>
<td>7.4</td>
<td>0</td>
<td>0</td>
<td>(0,12,13): 51.0</td>
</tr>
<tr>
<td>7</td>
<td>But a major outbreak in the Caribbean this year – with more than 100,000 cases reported – has health officials concerned.</td>
<td>16.4</td>
<td>1</td>
<td>0</td>
<td>(9,11,13): 50.4</td>
</tr>
<tr>
<td>8</td>
<td>Experts say American tourists are bringing Chikungunya back home, and it’s just a matter of time before it starts to spread within the United States.</td>
<td>10.6</td>
<td>0</td>
<td>0</td>
<td>(1,9,13): 50.1</td>
</tr>
<tr>
<td>9</td>
<td>After all, the Caribbean is a popular one with American tourists, and summer is fast approaching.</td>
<td>13.9</td>
<td>1</td>
<td>0</td>
<td>(12,13): 49.3</td>
</tr>
<tr>
<td>10</td>
<td>&quot;So far this year we’ve recorded eight travel-associated cases, and seven of them have come from countries in the Caribbean where we know the virus is being transmitted,&quot; Nasci said.</td>
<td>18.4</td>
<td>1</td>
<td>0</td>
<td>(7,11,13): 47.8</td>
</tr>
<tr>
<td>11</td>
<td>Other states have also reported cases of Chikungunya.</td>
<td>13.4</td>
<td>0</td>
<td>1</td>
<td>(9,13): 47.5</td>
</tr>
<tr>
<td>12</td>
<td>The Tennessee Department of Health said the state has had multiple cases of the virus in people who have traveled to the Caribbean.</td>
<td>15.6</td>
<td>1</td>
<td>0</td>
<td>(1,9,13): 46.9</td>
</tr>
<tr>
<td>13</td>
<td>The virus is not deadly, but it can be painful, with symptoms lasting for weeks.</td>
<td>54.5</td>
<td>1</td>
<td>1</td>
<td>(3,11,13): 45.5</td>
</tr>
<tr>
<td>14</td>
<td>Those with weak immune systems, such as the elderly, are more likely to suffer from the virus’ side effects than those who are healthier.</td>
<td>5.5</td>
<td>0</td>
<td>0</td>
<td>(4,11,13): 45.3</td>
</tr>
</tbody>
</table>

Table 1: An abridged CNN article (only first 15 out of 31 sentences are shown) and its “story highlights”. The latter are typically written by journalists to allow readers to quickly gather information on stories. Highlights are often used as gold standard abstractive summaries in the summarization literature.

Taint to the few sentences deemed most suitable to form the summary. However, a model trained with cross-entropy loss on collective labels will underfit the data as it will only maximize probabilities $p(1|s_i, D, \theta)$ for sentences in this set (e.g., sentences $\{0,11,13\}$ in Table 1) and ignore all other sentences. We found that there are many candidate summaries with high ROUGE scores which could be considered during training.

Table 1 (last column) shows candidate summaries ranked according to the mean of ROUGE-1, ROUGE-2, and ROUGE-L $F_1$ scores. Interestingly, multiple top ranked summaries have reasonably high ROUGE scores. For example, the average ROUGE for the summaries ranked second $(0,13)$, third $(11,13)$, and fourth $(0,11,13)$ is $57.5\%$, $57.2\%$, and $57.1\%$, and all top 16 summaries have ROUGE scores more or equal to $50\%$. A few sentences are indicative of important content and appear frequently in the summaries: sentence 13 occurs in all summaries except one, while sentence 0 appears in several summaries too. Also note that summaries $(11,13)$ and $(1,13)$ yield better ROUGE scores compared to longer summaries, and may be as informative, yet more concise, alternatives.

These discrepancies render the model less efficient at ranking sentences for the summarization task. Instead of maximizing the likelihood of the ground-truth labels, we could train the model to predict the individual ROUGE score for each sentence in the document and then select the top $m$ sentences with highest scores. But sentences with individual ROUGE scores do not necessarily lead to a high scoring summary, e.g., they may convey overlapping content and form verbose and redundant summaries. For example, sentence 3, despite having a high individual ROUGE score (35.6%), does not occur in any of the top 5 summaries. We next explain how we address these issues using reinforcement learning.

## 4 Sentence Ranking with Reinforcement Learning

Reinforcement learning (Sutton and Barto, 1998) has been proposed as a way of training sequence-to-sequence generation models in order to directly optimize the metric used at test time, e.g., BLEU or ROUGE (Ranzato et al., 2015). We adapt re-
inforcement learning to our formulation of extractive summarization to rank sentences for summary generation. We propose an objective function that combines the maximum-likelihood cross-entropy loss with rewards from policy gradient reinforcement learning to globally optimize ROUGE. Our training algorithm allows to explore the space of possible summaries, making our model more robust to unseen data. As a result, reinforcement learning helps extractive summarization in two ways: (a) it directly optimizes the evaluation metric instead of maximizing the likelihood of the ground-truth labels and (b) it makes our model better at discriminating among sentences; a sentence is ranked high for selection if it often occurs in high scoring summaries.

4.1 Policy Learning

We cast the neural summarization model introduced in Figure 1 in the Reinforcement Learning paradigm (Sutton and Barto, 1998). Accordingly, the model can be viewed as an “agent” which interacts with an “environment” consisting of documents. At first, the agent is initialized randomly, it reads document D and predicts a relevance score for each sentence \( s_i \in D \) using “policy” \( p(y_i|s_i, D, \theta) \), where \( \theta \) are model parameters. Once the agent is done reading the document, a summary with labels \( \hat{y} \) is sampled out of the ranked sentences. The agent is then given a “reward” \( r \) commensurate with how well the extract resembles the gold-standard summary. Specifically, as reward function we use mean F-

\[
L(\theta) = -\mathbb{E}_{\hat{y} \sim p_{\theta}} [r(\hat{y})] 
\]

where, \( p_{\theta} \) stands for \( p(\hat{y}|D, \theta) \). REINFORCE is based on the observation that the expected gradient of a non-differentiable reward function (ROUGE, in our case) can be computed as follows:

\[
\nabla L(\theta) = -\mathbb{E}_{\hat{y} \sim p_{\theta}} [r(\hat{y}) \nabla \log p(\hat{y}|D, \theta)] 
\]

While MLE in Equation (1) aims to maximize the likelihood of the training data, the objective in Equation (2) learns to discriminate among sentences with respect to how often they occur in high scoring summaries.

4.2 Training with High Probability Samples

Computing the expectation term in Equation (3) is prohibitive, since there is a large number of possible extracts. In practice, we approximate the expected gradient using a single sample \( \hat{y} \) from \( p_{\theta} \) for each training example in a batch:

\[
\nabla L(\theta) \approx -r(\hat{y}) \nabla \log p(\hat{y}|D, \theta) 
\]

Presented in its original form, the REINFORCE algorithm starts learning with a random policy which can make model training challenging for complex tasks like ours where a single document can give rise to a very large number of candidate summaries. We therefore limit the search space of \( \hat{y} \) in Equation (4) to the set of largest probability samples \( \hat{Y} \). We approximate \( \hat{Y} \) by the \( k \) extracts which receive highest ROUGE scores. More concretely, we assemble candidate summaries efficiently by first selecting \( p \) sentences from the document which on their own have high ROUGE scores. We then generate all possible combinations of \( p \) sentences subject to maximum length \( m \) and evaluate them against the gold summary. Summaries are ranked according to \( F_1 \) by taking the mean of ROUGE-1, ROUGE-2, and ROUGE-L. \( \hat{Y} \) contains these top \( k \) candidate summaries. During training, we sample \( \hat{y} \) from \( \hat{Y} \) instead of \( p(\hat{y}|D, \theta) \).

Ranzato et al. (2015) proposed an alternative to REINFORCE called MIXER (Mixed Incremental Cross-Entropy Reinforce) which first pretrains the model with the cross-entropy loss using ground truth labels and then follows a curriculum learning strategy (Bengio et al., 2015) to gradually teach the model to produce stable predictions on its own. In our experiments MIXER performed worse than the model of Nallapati et al. (2017) just trained on collective labels. We conjecture that this is due to the unbounded nature of our ranking problem. Recall that our model assigns relevance scores to sentences rather than words. The space of sentential representations is vast and fairly unconstrained compared to other prediction tasks operating with fixed vocabularies (Li et al., 2016; Paulus et al., 2017; Zhang and Lapata, 2017). Moreover, our approximation of the gradient allows the model to converge much faster to an optimal policy. Advantageously, we do not require an online reward estimator, we pre-compute \( \hat{Y} \), which leads to a significant speedup during training compared to MIXER.
(Ranzato et al., 2015) and related training schemes (Shen et al., 2016).

5 Experimental Setup

In this section we present our experimental setup for assessing the performance of our model which we call REFRESH as a shorthand for REInFoRcement Learning-based Extractive Summarization. We describe our datasets, discuss implementation details, our evaluation protocol, and the systems used for comparison.

Summarization Datasets We evaluated our models on the CNN and DailyMail news highlights datasets (Hermann et al., 2015). 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). We did not anonymize entities or lower case tokens. We followed previous studies (Cheng and Lapata, 2016; Nallapati et al., 2016, 2017; See et al., 2017; Tan and Wan, 2017) in assuming that the “story highlights” associated with each article are gold-standard abstractive summaries. During training we use these to generate high scoring extracts and to estimate rewards for them, but during testing, they are used as reference summaries to evaluate our models.

Implementation Details We generated extracts by selecting three sentences (m = 3) for CNN articles and four sentences (m = 4) for DailyMail articles. These decisions were informed by the fact that gold highlights in the CNN/DailyMail validation sets are 2.6/4.2 sentences long. For both datasets, we estimated high-scoring extracts using 10 document sentences (p = 10) with highest ROUGE scores. We tuned the initialization parameter k for \( \gamma \) on the validation set: we found that our model performs best with \( k = 5 \) for the CNN dataset and \( k = 15 \) for the DailyMail dataset.

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. Known words were initialized with pre-trained embeddings of size 200. Embeddings for unknown words were initialized to zero, but estimated during training. Sentences were padded with zeros to a length of 100. For the sentence encoder, we used a list of kernels of widths 1 to 7, each with output channel size of 50 (Kim et al., 2016). The sentence embedding size in our model was 350.

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. Known words were initialized with pre-trained embeddings of size 200. Embeddings for unknown words were initialized to zero, but estimated during training. Sentences were padded with zeros to a length of 100. For the sentence encoder, we used a list of kernels of widths 1 to 7, each with output channel size of 50 (Kim et al., 2016). The sentence embedding size in our model was 350.

For the recurrent neural network component in the document encoder and sentence extractor, we used a single-layered LSTM network with size 600. All input documents were padded with zeros to a maximum document length of 120. We performed minibatch cross-entropy training with a batch size of 20 documents for 20 training epochs. It took around 12 hrs on a single GPU to train. After each epoch, we evaluated our model on the validation set and chose the best performing model for the test set. During training we used the Adam optimizer (Kingma and Ba, 2015) with initial learning rate 0.001. Our system is implemented in TensorFlow (Abadi et al., 2015).

Evaluation We evaluated summarization quality using F1 ROUGE (Lin and Hovy, 2003). We report unigram and bigram overlap (ROUGE-1 and ROUGE-2) as a means of assessing informativeness and the longest common subsequence (ROUGE-L) as a means of assessing fluency.\(^2\) We compared REFRESH against a baseline which simply selects the first \( m \) leading sentences from each

\(^2\)We used pyrouge, a Python package, to compute all ROUGE scores with parameters "-a -c 85 -m -n 4 -w 1.2."
Our second experiment assessed the degree to which our model retains key information from the document following a question-answering (QA) paradigm which has been previously used to evaluate summary quality and text compression (Morris et al., 1992; Mani et al., 2002; Clarke and Lapata, 2010). We created a set of questions based on the gold summary under the assumption that it highlights the most important document content. We then examined whether participants were able to answer these questions by reading system summaries alone without access to the article. The more questions a system can answer, the better it is at summarizing the document as a whole.

We worked on the same 20 documents used in our first elicitation study. We wrote multiple fact-based question-answer pairs for each gold summary without looking at the document. Questions were formulated so as to not reveal answers to subsequent questions. We created 71 questions in total varying from two to six questions per gold summary. Example questions are given in Figure 2. Participants read the summary and answered all associated questions as best they could without access to the original document or the gold summary. Subjects were shown summaries from three systems: the LEAD baseline, the abstractive system of See et al. (2017), and Refresh. Five participants answered questions for each summary. We used the same scoring mechanism from Clarke and Lapata (2010), i.e., a correct answer was marked with a score of one, partially correct answers with a score of 0.5, and zero otherwise. The final score for a system is the average of all its question scores. Answers were elicited using Amazon’s Mechanical Turk crowdsourcing platform. We uploaded data in batches (one system at a time) on Mechanical Turk to ensure that same participant does not evaluate summaries from different systems on the same set of questions.

6 Results

We report results using automatic metrics in Table 2. The top part of the table compares Refresh against related extractive systems. The bottom part reports the performance of abstractive systems. We present three variants of LEAD, one is computed by ourselves and the other two are reported in Nallapati et al. (2017) and See et al. (2017). Note that they vary slightly due to differences in the preprocessing of the data. We report results on the CNN and DailyMail datasets and their combination (CNN+DailyMail).

Cross-Entropy vs Reinforcement Learning

The results in Table 2 show that Refresh is superior to our LEAD baseline and extractive systems across datasets and metrics. It outperforms the extractive system of Cheng and Lapata (2016) which is trained on individual labels. Refresh is not directly comparable with Nallapati et al. (2017) as they generate anonymized summaries. Their system lags behind their LEAD baseline on ROUGE-L on the CNN+DailyMail dataset (35.5% vs 35.3%). Also note that their model is trained
Extractive vs Abstractive Systems  Our automatic evaluation results further demonstrate that Refresh is superior to abstractive systems (Chen et al., 2016; Nallapati et al., 2016; See et al., 2017; Tan and Wan, 2017) which are all variants of an encoder-decoder architecture (Sutskever et al., 2014). Despite being more faithful to the actual summarization task (hand-written summaries combine several pieces of information from the original document), abstractive systems lag behind the LEAD baseline. Tan and Wan (2017) present a graph-based neural model, which manages to outperform LEAD on ROUGE-1 but falters when higher order ROUGE scores are used. Amongst abstractive systems See et al. (2017) perform best. Interestingly, their system is mostly extractive, exhibiting a small degree of rewriting; it copies more than 35% of the sentences in the source document, 85% of 4-grams, 90% of 3-grams, 95% of bigrams, and 99% of unigrams.

Human Evaluation: System Ranking Table 3 shows, proportionally, how often participants ranked each system, 1st, 2nd, and so on. Perhaps unsurprisingly human-authored summaries are considered best (and ranked 1st 39% of the time). Refresh is ranked 2nd best followed by LEAD and See et al. (2017) which are mostly ranked in 3rd and 4th places. We carried out pairwise comparisons between all models in Table 3 to assess whether system differences are statistically significant. There is no significant difference between LEAD and See et al. (2017), and Refresh and Gold (using a one-way ANOVA with post-hoc Tukey HSD tests; $p < 0.01$). All other differences are statistically significant.

Human Evaluation: Question Answering The results of our QA evaluation are shown in the last column of Table 3. Based on summaries generated by Refresh, participants can answer 66.34% of questions correctly. Summaries produced by LEAD and the abstractive system of See et al. (2017) provide answers for 36.33% and 28.73% of the questions, respectively. Differences between systems are all statistically significant ($p < 0.01$) with the exception of LEAD and See et al. (2017).

Although the QA results in Table 3 follow the same pattern as ROUGE in Table 2, differences among systems are now greatly amplified. QA-based evaluation is more focused and a closer reflection of users’ information need (i.e., to find out what the article is about), whereas ROUGE simply captures surface similarity (i.e., n-gram overlap) between output summaries and their references. Interestingly, LEAD is considered better than See et al. (2017) in the QA evaluation, whereas we find the opposite when participants are asked to rank systems. We hypothesize that LEAD is indeed more informative than See et al. (2017) but...
humans prefer shorter summaries. The average length of LEAD summaries is 105.7 words compared to 61.6 for See et al. (2017).

7 Related Work

Traditional summarization methods manually define features to rank sentences for their salience in order to identify the most important sentences in a document or set of documents (Kupiec et al., 1995; Mani, 2001; Radev et al., 2004; Filatova and Hatzivassiloglou, 2004; Nenkova et al., 2006; Spärck Jones, 2007). A vast majority of these methods learn to score each sentence independently (Barzilay and Elhadad, 1997; Teufel and Moens, 1997; Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Shen et al., 2007; Schilder and Kondadadi, 2008; Wan, 2010) and a summary is generated by selecting top-scored sentences in a way that is not incorporated into the learning process. Summary quality can be improved heuristically, (Yih et al., 2007), via max-margin methods (Carbonell and Goldstein, 1998; Li et al., 2009), or integer-linear programming (Woodsend and Lapata, 2010; Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012; Almeida and Martins, 2013; Parveen et al., 2015).

Recent deep learning methods (Kågebäck et al., 2014; Yin and Pei, 2015; Cheng and Lapata, 2016; Nallapati et al., 2017) learn continuous features without any linguistic preprocessing (e.g., named entities). Like traditional methods, these approaches also suffer from the mismatch between the learning objective and the evaluation criterion (e.g., ROUGE) used at the test time. In comparison, our neural model globally optimizes the ROUGE evaluation metric through a reinforcement learning objective: sentences are highly ranked if they occur in highly scoring summaries.

Reinforcement learning has been previously used in the context of traditional multi-document summarization as a means of selecting a sentence or a subset of sentences from a document cluster. Ryang and Abekawa (2012) cast the sentence selection task as a search problem. Their agent observes a state (e.g., a candidate summary), executes an action (a transition operation that produces a new state selecting a not-yet-selected sentence), and then receives a delayed reward based on tf*idf. Follow-on work (Rioux et al., 2014) extends this approach by employing ROUGE as part of the reward function, while Henß et al. (2015) further experiment with Q-learning. Mollá-Aliod (2017) has adapt this approach to query-focused summarization. Our model differs from these approaches both in application and formulation. We focus solely on extractive summarization, in our case states are documents (not summaries) and actions are relevance scores which lead to sentence ranking (not sentence-to-sentence transitions). Rather than employing reinforcement learning for sentence selection, our algorithm performs sentence ranking using ROUGE as the reward function.

The REINFORCE algorithm (Williams, 1992) has been shown to improve encoder-decoder text-rewriting systems by allowing to directly optimize a non-differentiable objective (Ranzato et al., 2015; Li et al., 2016; Paulus et al., 2017) or to inject task-specific constraints (Zhang and Lapata, 2017; Nogueira and Cho, 2017). However, we are not aware of any attempts to use reinforcement learning for training a sentence ranker in the context of extractive summarization.

8 Conclusions

In this work we developed an extractive summarization model which is globally trained by optimizing the ROUGE evaluation metric. Our training algorithm explores the space of candidate summaries while learning to optimize a reward function which is relevant for the task at hand. Experimental results show that reinforcement learning offers a great means to steer our model towards generating informative, fluent, and concise summaries outperforming state-of-the-art extractive and abstractive systems on the CNN and DailyMail datasets. In the future we would like to focus on smaller discourse units (Mann and Thompson, 1988) rather than individual sentences, modeling compression and extraction jointly.

Acknowledgments We gratefully acknowledge the support of the European Research Council (Lapata; award number 681760), the European Union under the Horizon 2020 SUMMA project (Narayan, Cohen; grant agreement 688139), and Huawei Technologies (Cohen). The research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract FA8650-17-C-9118. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.
References


