A Neural Attention Model for Sentence Summarization

Alexander M. Rush
Facebook AI Research /
Harvard SEAS
srush@seas.harvard.edu

Sumit Chopra
Facebook AI Research
spchopra@fb.com

Jason Weston
Facebook AI Research
jase@fb.com

Abstract

Summarization based on text extraction is inherently limited, but generation-style abstractive methods have proven challenging to build. In this work, we propose a fully data-driven approach to abstractive sentence summarization. Our method utilizes a local attention-based model that generates each word of the summary conditioned on the input sentence. While the model is structurally simple, it can easily be trained end-to-end and scales to a large amount of training data. The model shows significant performance gains on the DUC-2004 shared task compared with several strong baselines.

1 Introduction

Summarization is an important challenge of natural language understanding. The aim is to produce a condensed representation of an input text that captures the core meaning of the original. Most successful summarization systems utilize *extractive* approaches that crop out and stitch together portions of the text to produce a condensed version. In contrast, *abstractive* summarization attempts to produce a bottom-up summary, aspects of which may not appear as part of the original.

We focus on the task of sentence-level summarization. While much work on this task has looked at deletion-based sentence compression techniques (Knight and Marcu (2002), among many others), studies of human summarizers show that it is common to apply various other operations while condensing, such as paraphrasing, generalization, and reordering (Jing, 2002). Past work has modeled this abstractive summarization problem either using linguistically-inspired constraints (Dorr et al., 2003; Zajic et al., 2004) or with syntactic transformations of the input text (Cohn and Lapata, 2008; Woodsend et al., 2010). These approaches are described in more detail in Section 6.

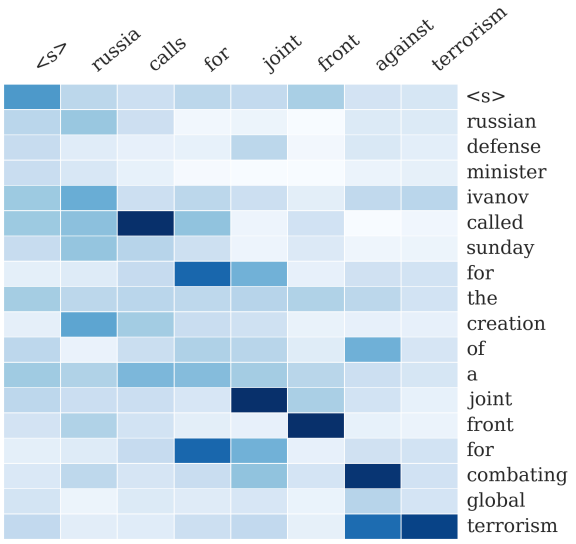

Figure 1: Example output of the attention-based summarization (ABS) system. The heatmap represents a soft alignment between the input (right) and the generated summary (top). The columns represent the distribution over the input after generating each word.

We instead explore a fully data-driven approach for generating abstractive summaries. Inspired by the recent success of neural machine translation, we combine a neural language model with a contextual input encoder. Our encoder is modeled off of the attention-based encoder of Bahdanau et al. (2014) in that it learns a latent soft alignment over the input text to help inform the summary (as shown in Figure 1). Crucially both the encoder and the generation model are trained jointly on the sentence summarization task. The model is described in detail in Section 3. Our model also incorporates a beam-search decoder as well as additional features to model extractive elements; these aspects are discussed in Sections 4 and 5.

This approach to summarization, which we call Attention-Based Summarization (ABS), incorporates less linguistic structure than comparable abstractive summarization approaches, but can easily

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 379–389,
Lisbon, Portugal, 17-21 September 2015.

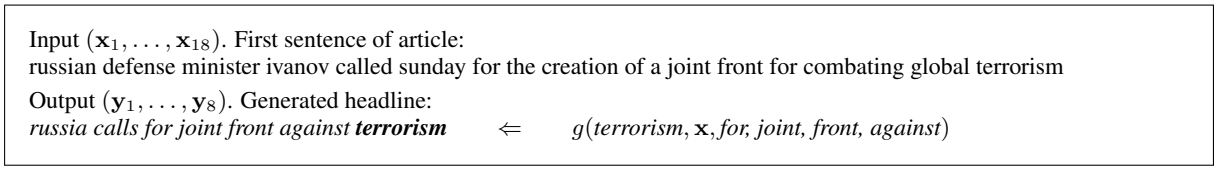

Figure 2: Example input sentence and the generated summary. The score of generating $\mathbf{y}_{i+1}$ (`terrorism`) is based on the context $\mathbf{y}_c$ (`for ... against`) as well as the input $\mathbf{x}_1 \ldots \mathbf{x}_{18}$. Note that the summary generated is abstractive which makes it possible to *generalize* (`russian defense minister` to `russia`) and *paraphrase* (`for combating` to `against`), in addition to *compressing* (dropping `the creation of`), see Jing (2002) for a survey of these editing operations.

scale to train on a large amount of data. Since our system makes no assumptions about the vocabulary of the generated summary it can be trained directly on any document-summary pair.[1] This allows us to train a summarization model for headline-generation on a corpus of article pairs from Gigaword (Graff et al., 2003) consisting of around 4 million articles. An example of generation is given in Figure 2, and we discuss the details of this task in Section 7.

To test the effectiveness of this approach we run extensive comparisons with multiple abstractive and extractive baselines, including traditional syntax-based systems, integer linear program-constrained systems, information-retrieval style approaches, as well as statistical phrase-based machine translation. Section 8 describes the results of these experiments. Our approach outperforms a machine translation system trained on the same large-scale dataset and yields a large improvement over the highest scoring system in the DUC-2004 competition.

2 Background

We begin by defining the sentence summarization task. Given an input sentence, the goal is to produce a condensed summary. Let the input consist of a sequence of M words $\mathbf{x}_1, \ldots, \mathbf{x}_M$ coming from a fixed vocabulary $\mathcal{V}$ of size $|\mathcal{V}| = V$. We will represent each word as an indicator vector $\mathbf{x}_i \in \{0,1\}^V$ for $i \in \{1, \ldots, M\}$, sentences as a sequence of indicators, and $\mathcal{X}$ as the set of possible inputs. Furthermore define the notation $\mathbf{x}_{[i,j,k]}$ to indicate the sub-sequence of elements i, j, k.

A summarizer takes $\mathbf{x}$ as input and outputs a shortened sentence $\mathbf{y}$ of length $N < M$. We will assume that the words in the summary also come from the same vocabulary $\mathcal{V}$ and that the output is a sequence $\mathbf{y}_1, \ldots, \mathbf{y}_N$. Note that in contrast to related tasks, like machine translation, we will assume that the output length N is fixed, and that the system knows the length of the summary before generation.[2]

Next consider the problem of generating summaries. Define the set $\mathcal{Y} \subset (\{0,1\}^V, \ldots, \{0,1\}^V)$ as all possible sentences of length N, i.e. for all i and $\mathbf{y} \in \mathcal{Y}$, $\mathbf{y}_i$ is an indicator. We say a system is *abstractive* if it tries to find the optimal sequence from this set $\mathcal{Y}$,

$$\arg\max_{\mathbf{y} \in \mathcal{Y}} s(\mathbf{x}, \mathbf{y}), \tag{1}$$

under a scoring function $s : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$. Contrast this to a fully *extractive* sentence summary[3] which transfers words from the input:

$$\arg\max_{m \in \{1,\ldots M\}^N} s(\mathbf{x}, \mathbf{x}_{[m_1,\ldots,m_N]}), \tag{2}$$

or to the related problem of sentence *compression* that concentrates on deleting words from the input:

$$\arg\max_{m \in \{1,\ldots M\}^N, m_{i-1} < m_i} s(\mathbf{x}, \mathbf{x}_{[m_1,\ldots,m_N]}). \tag{3}$$

While abstractive summarization poses a more difficult generation challenge, the lack of hard constraints gives the system more freedom in generation and allows it to fit with a wider range of training data.

In this work we focus on factored scoring functions, s, that take into account a fixed window of previous words:

$$s(\mathbf{x}, \mathbf{y}) \approx \sum_{i=0}^{N-1} g(\mathbf{y}_{i+1}, \mathbf{x}, \mathbf{y}_c), \tag{4}$$

[1] In contrast to a large-scale sentence compression systems like Filippova and Altun (2013) which require monotonic aligned compressions.

[2] For the DUC-2004 evaluation, it is actually the number of *bytes* of the output that is capped. More detail is given in Section 7.

[3] Unfortunately the literature is inconsistent on the formal definition of this distinction. Some systems self-described as abstractive would be extractive under our definition.

where we define $\mathbf{y}_c \triangleq \mathbf{y}_{[i-C+1,\ldots,i]}$ for a window of size C.

In particular consider the conditional log-probability of a summary given the input, $s(\mathbf{x}, \mathbf{y}) = \log p(\mathbf{y}|\mathbf{x}; \theta)$. We can write this as:

$$\log p(\mathbf{y}|\mathbf{x}; \theta) \approx \sum_{i=0}^{N-1} \log p(\mathbf{y}_{i+1}|\mathbf{x}, \mathbf{y}_c; \theta),$$

where we make a Markov assumption on the length of the context as size C and assume for $i < 1$, $\mathbf{y}_i$ is a special start symbol $\langle S \rangle$.

With this scoring function in mind, our main focus will be on modelling the local conditional distribution: $p(\mathbf{y}_{i+1}|\mathbf{x}, \mathbf{y}_c; \theta)$. The next section defines a parameterization for this distribution, in Section 4, we return to the question of generation for factored models, and in Section 5 we introduce a modified factored scoring function.

3 Model

The distribution of interest, $p(\mathbf{y}_{i+1}|\mathbf{x}, \mathbf{y}_c; \theta)$, is a conditional language model based on the input sentence $\mathbf{x}$. Past work on summarization and compression has used a noisy-channel approach to split and independently estimate a language model and a conditional summarization model (Banko et al., 2000; Knight and Marcu, 2002; Daumé III and Marcu, 2002), i.e.,

$$\arg\max_{\mathbf{y}} \log p(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{y}} \log p(\mathbf{y}) p(\mathbf{x}|\mathbf{y})$$

where $p(y)$ and $p(x|y)$ are estimated separately. Here we instead follow work in neural machine translation and directly parameterize the original distribution as a neural network. The network contains both a neural probabilistic language model and an encoder which acts as a conditional summarization model.

3.1 Neural Language Model

The core of our parameterization is a language model for estimating the contextual probability of the next word. The language model is adapted from a standard feed-forward neural network language model (NNLM), particularly the class of NNLMs described by Bengio et al. (2003). The full model is:

$$\begin{aligned} p(\mathbf{y}_{i+1}|\mathbf{y}_c, \mathbf{x}; \theta) &\propto \exp(\mathbf{V}\mathbf{h} + \mathbf{W}\mathrm{enc}(\mathbf{x}, \mathbf{y}_c)), \\ \tilde{\mathbf{y}}_c &= [\mathbf{E}\mathbf{y}_{i-C+1}, \ldots, \mathbf{E}\mathbf{y}_i], \\ \mathbf{h} &= \tanh(\mathbf{U}\tilde{\mathbf{y}}_c). \end{aligned}$$

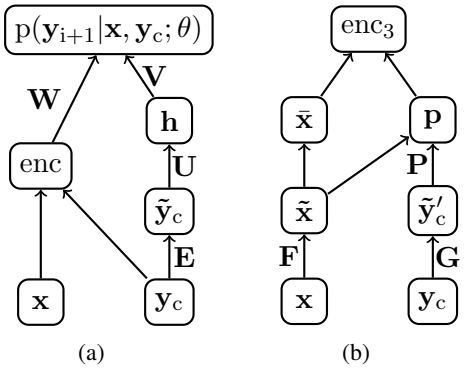

Figure 3: (a) A network diagram for the NNLM decoder with additional encoder element. (b) A network diagram for the attention-based encoder enc_3.

The parameters are $\theta = (\mathbf{E}, \mathbf{U}, \mathbf{V}, \mathbf{W})$ where $\mathbf{E} \in \mathbb{R}^{D \times V}$ is a word embedding matrix, $\mathbf{U} \in \mathbb{R}^{(CD) \times H}$, $\mathbf{V} \in \mathbb{R}^{V \times H}$, $\mathbf{W} \in \mathbb{R}^{V \times H}$ are weight matrices,[4] D is the size of the word embeddings, and $\mathbf{h}$ is a hidden layer of size H. The black-box function enc is a contextual encoder term that returns a vector of size H representing the input and current context; we consider several possible variants, described subsequently. Figure 3a gives a schematic representation of the decoder architecture.

3.2 Encoders

Note that without the encoder term this represents a standard language model. By incorporating in enc and training the two elements jointly we crucially can incorporate the input text into generation. We discuss next several possible instantiations of the encoder.

Bag-of-Words Encoder Our most basic model simply uses the bag-of-words of the input sentence embedded down to size H, while ignoring properties of the original order or relationships between neighboring words. We write this model as:

$$\begin{aligned} \mathrm{enc}_1(\mathbf{x}, \mathbf{y}_c) &= \mathbf{p}^\top \tilde{\mathbf{x}}, \\ \mathbf{p} &= [1/M, \ldots, 1/M], \\ \tilde{\mathbf{x}} &= [\mathbf{F}\mathbf{x}_1, \ldots, \mathbf{F}\mathbf{x}_M]. \end{aligned}$$

Where the input-side embedding matrix $\mathbf{F} \in \mathbb{R}^{H \times V}$ is the only new parameter of the encoder and $\mathbf{p} \in [0, 1]^M$ is a uniform distribution over the input words.

[4] Each of the weight matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ also has a corresponding bias term. For readability, we omit these terms throughout the paper.

For summarization this model can capture the relative importance of words to distinguish content words from stop words or embellishments. Potentially the model can also learn to combine words; although it is inherently limited in representing contiguous phrases.

Convolutional Encoder To address some of the modelling issues with bag-of-words we also consider using a deep convolutional encoder for the input sentence. This architecture improves on the bag-of-words model by allowing local interactions between words while also not requiring the context $\mathbf{y}_{\mathrm{c}}$ while encoding the input.

We utilize a standard time-delay neural network (TDNN) architecture, alternating between temporal convolution layers and max pooling layers.

$$\forall j,\ \mathrm{enc}_2(\mathbf{x}, \mathbf{y}_{\mathrm{c}})_j = \max_i \tilde{\mathbf{x}}^L_{i,j}, \quad (5)$$

$$\forall i, l \in \{1, \dots L\},\ \tilde{\mathbf{x}}^l_j = \tanh(\max\{\bar{\mathbf{x}}^l_{2i-1}, \bar{\mathbf{x}}^l_{2i}\}), \quad (6)$$

$$\forall i, l \in \{1, \dots L\},\ \bar{\mathbf{x}}^l_i = \mathbf{Q}^l \tilde{\mathbf{x}}^{l-1}_{[i-Q,\dots,i+Q]}, \quad (7)$$

$$\tilde{\mathbf{x}}^0 = [\mathbf{F}\mathbf{x}_1, \dots, \mathbf{F}\mathbf{x}_M]. \quad (8)$$

Where $\mathbf{F}$ is a word embedding matrix and $\mathbf{Q}^{L \times H \times 2Q+1}$ consists of a set of filters for each layer $\{1, \dots L\}$. Eq. 7 is a temporal (1D) convolution layer, Eq. 6 consists of a 2-element temporal max pooling layer and a pointwise non-linearity, and final output Eq. 5 is a max over time. At each layer $\tilde{x}$ is one half the size of $\bar{x}$. For simplicity we assume that the convolution is padded at the boundaries, and that M is greater than 2^L so that the dimensions are well-defined.

Attention-Based Encoder While the convolutional encoder has richer capacity than bag-of-words, it still is required to produce a single representation for the entire input sentence. A similar issue in machine translation inspired Bahdanau et al. (2014) to instead utilize an attention-based contextual encoder that constructs a representation based on the generation context. Here we note that if we exploit this context, we can actually use a rather simple model similar to bag-of-words:

$$\begin{aligned}
\mathrm{enc}_3(\mathbf{x}, \mathbf{y}_{\mathrm{c}}) &= \mathbf{p}^\top \bar{\mathbf{x}}, \\
\mathbf{p} &\propto \exp(\tilde{\mathbf{x}} \mathbf{P} \tilde{\mathbf{y}}'_{\mathrm{c}}), \\
\tilde{\mathbf{x}} &= [\mathbf{F}\mathbf{x}_1, \dots, \mathbf{F}\mathbf{x}_M], \\
\tilde{\mathbf{y}}'_{\mathrm{c}} &= [\mathbf{G}\mathbf{y}_{i-C+1}, \dots, \mathbf{G}\mathbf{y}_i], \\
\forall i\ \ \bar{\mathbf{x}}_i &= \sum_{q=i-Q}^{i+Q} \tilde{\mathbf{x}}_i / Q.
\end{aligned}$$

Where $\mathbf{G} \in \mathbb{R}^{D \times V}$ is an embedding of the context, $\mathbf{P} \in \mathbb{R}^{H \times (CD)}$ is a new weight matrix parameter mapping between the context embedding and input embedding, and Q is a smoothing window. The full model is shown in Figure 3b.

Informally we can think of this model as simply replacing the uniform distribution in bag-of-words with a learned soft alignment, $\mathbf{P}$, between the input and the summary. Figure 1 shows an example of this distribution $\mathbf{p}$ as a summary is generated. The soft alignment is then used to weight the smoothed version of the input $\bar{\mathbf{x}}$ when constructing the representation. For instance if the current context aligns well with position i then the words $\mathbf{x}_{i-Q}, \dots, \mathbf{x}_{i+Q}$ are highly weighted by the encoder. Together with the NNLM, this model can be seen as a stripped-down version of the attention-based neural machine translation model.[5]

3.3 Training

The lack of generation constraints makes it possible to train the model on arbitrary input-output pairs. Once we have defined the local conditional model, $p(\mathbf{y}_{i+1} | \mathbf{x}, \mathbf{y}_{\mathrm{c}}; \theta)$, we can estimate the parameters to minimize the negative log-likelihood of a set of summaries. Define this training set as consisting of J input-summary pairs $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(J)}, \mathbf{y}^{(J)})$. The negative log-likelihood conveniently factors[6] into a term for each token in the summary:

$$\begin{aligned}
\mathrm{NLL}(\theta) &= -\sum_{j=1}^{J} \log p(\mathbf{y}^{(j)} | \mathbf{x}^{(j)}; \theta), \\
&= -\sum_{j=1}^{J} \sum_{i=1}^{N-1} \log p(\mathbf{y}^{(j)}_{i+1} | \mathbf{x}^{(j)}, \mathbf{y}_{\mathrm{c}}; \theta).
\end{aligned}$$

We minimize NLL by using mini-batch stochastic gradient descent. The details are described further in Section 7.

[5] To be explicit, compared to Bahdanau et al. (2014) our model uses an NNLM instead of a target-side LSTM, source-side windowed averaging instead of a source-side bi-directional RNN, and a weighted dot-product for alignment instead of an alignment MLP.

[6] This is dependent on using the gold standard contexts $\mathbf{y}_{\mathrm{c}}$. An alternative is to use the predicted context within a structured or reenforcement-learning style objective.

4 Generating Summaries

We now return to the problem of generating summaries. Recall from Eq. 4 that our goal is to find,

$$\mathbf{y}^* \quad = \quad \arg\max_{\mathbf{y} \in \mathcal{Y}} \sum_{i=0}^{N-1} g(\mathbf{y}_{i+1}, \mathbf{x}, \mathbf{y}_{\mathrm{c}}).$$

Unlike phrase-based machine translation where inference is NP-hard, it actually is tractable in theory to compute $\mathbf{y}^*$. Since there is no explicit hard alignment constraint, Viterbi decoding can be applied and requires $O(NV^C)$ time to find an exact solution. In practice though V is large enough to make this difficult.

An alternative approach is to approximate the $\arg\max$ with a strictly greedy or deterministic decoder. While decoders of this form can produce very bad approximations, they have shown to be relatively effective and fast for neural MT models (Sutskever et al., 2014).

A compromise between exact and greedy decoding is to use a beam-search decoder (Algorithm 1) which maintains the full vocabulary $\mathcal{V}$ while limiting itself to K potential hypotheses at each position of the summary. The beam-search algorithm is shown here:

Algorithm 1 Beam Search

Input: Parameters θ, beam size K, input $\mathbf{x}$
Output: Approx. K-best summaries

$\pi[0] \leftarrow \{\epsilon\}$
$\mathcal{S} = \mathcal{V}$ if abstractive else $\{\mathbf{x}_i \mid \forall i\}$
for $i = 0$ to $N-1$ **do**
  ▷ Generate Hypotheses
  $\mathcal{N} \leftarrow \left\{ [\mathbf{y}, \mathbf{y}_{i+1}] \quad | \quad \mathbf{y} \in \pi[i], \mathbf{y}_{i+1} \in \mathcal{S} \right\}$
  ▷ Hypothesis Recombination
  $\mathcal{H} \leftarrow \left\{ \mathbf{y} \in \mathcal{N} \quad | \quad \begin{array}{c} s(\mathbf{y}, \mathbf{x}) > s(\mathbf{y}', \mathbf{x}) \\ \forall \mathbf{y}' \in \mathcal{N} \text{ s.t. } \mathbf{y}_{\mathrm{c}} = \mathbf{y}'_{\mathrm{c}} \end{array} \right\}$
  ▷ Filter K-Max
  $\pi[i+1] \leftarrow \underset{\mathbf{y} \in \mathcal{H}}{\text{K-arg max}}\, g(\mathbf{y}_{i+1}, \mathbf{y}_{\mathrm{c}}, \mathbf{x}) + s(\mathbf{y}, \mathbf{x})$
end for
return $\pi[N]$

As with Viterbi this beam search algorithm is much simpler than beam search for phrase-based MT. Because there is no explicit constraint that each source word be used exactly once there is no need to maintain a bit set and we can simply move from left-to-right generating words. The beam search algorithm requires $O(KNV)$ time. From a computational perspective though, each round of beam search is dominated by computing $p(\mathbf{y}_i | \mathbf{x}, \mathbf{y}_{\mathrm{c}})$ for each of the K hypotheses. These can be computed as a mini-batch, which in practice greatly reduces the factor of K.

5 Extension: Extractive Tuning

While we will see that the attention-based model is effective at generating summaries, it does miss an important aspect seen in the human-generated references. In particular the abstractive model does not have the capacity to find extractive word matches when necessary, for example transferring unseen proper noun phrases from the input. Similar issues have also been observed in neural translation models particularly in terms of translating rare words (Luong et al., 2014).

To address this issue we experiment with tuning a very small set of additional features that tradeoff the abstractive/extractive tendency of the system. We do this by modifying our scoring function to directly estimate the probability of a summary using a log-linear model, as is standard in machine translation:

$$p(\mathbf{y} | \mathbf{x}; \theta, \alpha) \quad \propto \quad \exp(\alpha^\top \sum_{i=0}^{N-1} f(\mathbf{y}_{i+1}, \mathbf{x}, \mathbf{y}_{\mathrm{c}})).$$

Where $\alpha \in \mathbb{R}^5$ is a weight vector and f is a feature function. Finding the best summary under this distribution corresponds to maximizing a factored scoring function s,

$$s(\mathbf{y}, \mathbf{x}) \quad = \quad \sum_{i=0}^{N-1} \alpha^\top f(\mathbf{y}_{i+1}, \mathbf{x}, \mathbf{y}_{\mathrm{c}}).$$

where $g(\mathbf{y}_{i+1}, \mathbf{x}, \mathbf{y}_{\mathrm{c}}) \triangleq \alpha^\top f(\mathbf{y}_{i+1}, \mathbf{x}, \mathbf{y}_{\mathrm{c}})$ to satisfy Eq. 4. The function f is defined to combine the local conditional probability with some additional indicator featrues:

$$\begin{aligned} f(\mathbf{y}_{i+1}, \mathbf{x}, \mathbf{y}_{\mathrm{c}}) = [\ & \log p(\mathbf{y}_{i+1} | \mathbf{x}, \mathbf{y}_{\mathrm{c}}; \theta), \\ & \mathbb{1}\{\exists j.\ \mathbf{y}_{i+1} = \mathbf{x}_j\ \}, \\ & \mathbb{1}\{\exists j.\ \mathbf{y}_{i+1-k} = \mathbf{x}_{j-k}\ \forall k \in \{0, 1\}\}, \\ & \mathbb{1}\{\exists j.\ \mathbf{y}_{i+1-k} = \mathbf{x}_{j-k}\ \forall k \in \{0, 1, 2\}\}, \\ & \mathbb{1}\{\exists k > j.\ \mathbf{y}_i = \mathbf{x}_k, \mathbf{y}_{i+1} = \mathbf{x}_j\}\ ]. \end{aligned}$$

These features correspond to indicators of unigram, bigram, and trigram match with the input as well as reordering of input words. Note that setting $\alpha = \langle 1, 0, \ldots, 0 \rangle$ gives a model identical to standard ABS.

After training the main neural model, we fix θ and tune the α parameters. We follow the statistical machine translation setup and use minimum-error rate training (MERT) to tune for the summarization metric on tuning data (Och, 2003). This tuning step is also identical to the one used for the phrase-based machine translation baseline.

6 Related Work

Abstractive sentence summarization has been traditionally connected to the task of headline generation. Our work is similar to early work of Banko et al. (2000) who developed a statistical machine translation-inspired approach for this task using a corpus of headline-article pairs. We extend this approach by: (1) using a neural summarization model as opposed to a count-based noisy-channel model, (2) training the model on much larger scale (25K compared to 4 million articles), (3) and allowing fully abstractive decoding.

This task was standardized around the DUC-2003 and DUC-2004 competitions (Over et al., 2007). The TOPIARY system (Zajic et al., 2004) performed the best in this task, and is described in detail in the next section. We point interested readers to the DUC web page (`http://duc.nist.gov/`) for the full list of systems entered in this shared task.

More recently, Cohn and Lapata (2008) give a compression method which allows for more arbitrary transformations. They extract tree transduction rules from aligned, parsed texts and learn weights on transfomations using a max-margin learning algorithm. Woodsend et al. (2010) propose a quasi-synchronous grammar approach utilizing both context-free parses and dependency parses to produce legible summaries. Both of these approaches differ from ours in that they directly use the syntax of the input/output sentences. The latter system is W&L in our results; we attempted to train the former system T3 on this dataset but could not train it at scale.

In addition to Banko et al. (2000) there has been some work using statistical machine translation directly for abstractive summary. Wubben et al. (2012) utilize MOSES directly as a method for text simplification.

Recently Filippova and Altun (2013) developed a strictly extractive system that is trained on a relatively large corpora (250K sentences) of article-title pairs. Because their focus is extractive compression, the sentences are transformed by a series of heuristics such that the words are in monotonic alignment. Our system does not require this alignment step but instead uses the text directly.

Neural MT This work is closely related to recent work on neural network language models (NNLM) and to work on neural machine translation. The core of our model is a NNLM based on that of Bengio et al. (2003).

Recently, there have been several papers about models for machine translation (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014). Of these our model is most closely related to the attention-based model of Bahdanau et al. (2014), which explicitly finds a soft alignment between the current position and the input source. Most of these models utilize recurrent neural networks (RNNs) for generation as opposed to feed-forward models. We hope to incorporate an RNN-LM in future work.

7 Experimental Setup

We experiment with our attention-based sentence summarization model on the task of headline generation. In this section we describe the corpora used for this task, the baseline methods we compare with, and implementation details of our approach.

7.1 Data Set

The standard sentence summarization evaluation set is associated with the DUC-2003 and DUC-2004 shared tasks (Over et al., 2007). The data for this task consists of 500 news articles from the New York Times and Associated Press Wire services each paired with 4 different human-generated reference summaries (not actually headlines), capped at 75 bytes. This data set is evaluation-only, although the similarly sized DUC-2003 data set was made available for the task. The expectation is for a summary of roughly 14 words, based on the text of a complete article (although we only make use of the first sentence). The full data set is available by request at `http://duc.nist.gov/data.html`.

For this shared task, systems were entered and evaluated using several variants of the recall-oriented ROUGE metric (Lin, 2004). To make recall-only evaluation unbiased to length, output of all systems is cut-off after 75-characters and no bonus is given for shorter summaries.

Unlike BLEU which interpolates various n-gram matches, there are several versions of ROUGE for different match lengths. The DUC evaluation uses ROUGE-1 (unigrams), ROUGE-2 (bigrams), and ROUGE-L (longest-common substring), all of which we report.

In addition to the standard DUC-2014 evaluation, we also report evaluation on single reference headline-generation using a randomly held-out subset of Gigaword. This evaluation is closer to the task the model is trained for, and it allows us to use a bigger evaluation set, which we will include in our code release. For this evaluation, we tune systems to generate output of the average title length.

For training data for both tasks, we utilize the annotated Gigaword data set (Graff et al., 2003; Napoles et al., 2012), which consists of standard Gigaword, preprocessed with Stanford CoreNLP tools (Manning et al., 2014). Our model only uses annotations for tokenization and sentence separation, although several of the baselines use parsing and tagging as well. Gigaword contains around 9.5 million news articles sourced from various domestic and international news services over the last two decades.

For our training set, we pair the headline of each article with its first sentence to create an input-summary pair. While the model could in theory be trained on any pair, Gigaword contains many spurious headline-article pairs. We therefore prune training based on the following heuristic filters: (1) Are there no non-stop-words in common? (2) Does the title contain a byline or other extraneous editing marks? (3) Does the title have a question mark or colon? After applying these filters, the training set consists of roughly $J = 4$ million title-article pairs. We apply a minimal preprocessing step using PTB tokenization, lower-casing, replacing all digit characters with #, and replacing of word types seen less than 5 times with UNK. We also remove all articles from the time-period of the DUC evaluation. release.

The complete input training vocabulary consists of 119 million word tokens and 110K unique word types with an average sentence size of 31.3 words. The headline vocabulary consists of 31 million tokens and 69K word types with the average title of length 8.3 words (note that this is significantly shorter than the DUC summaries). On average there are 4.6 overlapping word types between the headline and the input; although only 2.6 in the first 75-characters of the input.

7.2 Baselines

Due to the variety of approaches to the sentence summarization problem, we report a broad set of headline-generation baselines.

From the DUC-2004 task we include the PREFIX baseline that simply returns the first 75-characters of the input as the headline. We also report the winning system on this shared task, TOPIARY (Zajic et al., 2004). TOPIARY merges a compression system using linguistically-motivated transformations of the input (Dorr et al., 2003) with an unsupervised topic detection (UTD) algorithm that appends key phrases from the full article onto the compressed output. Woodsend et al. (2010) (described above) also report results on the DUC dataset.

The DUC task also includes a set of manual summaries performed by 8 human summarizers each summarizing half of the test data sentences (yielding 4 references per sentence). We report the average inter-annotater agreement score as REFERENCE. For reference, the best human evaluator scores 31.7 ROUGE-1.

We also include several baselines that have access to the same training data as our system. The first is a sentence compression baseline COMPRESS (Clarke and Lapata, 2008). This model uses the syntactic structure of the original sentence along with a language model trained on the headline data to produce a compressed output. The syntax and language model are combined with a set of linguistic constraints and decoding is performed with an ILP solver.

To control for memorizing titles from training, we implement an information retrieval baseline, IR. This baseline indexes the training set, and gives the title for the article with highest BM-25 match to the input (see Manning et al. (2008)).

Finally, we use a phrase-based statistical machine translation system trained on Gigaword to produce summaries, MOSES+ (Koehn et al., 2007). To improve the baseline for this task, we augment the phrase table with "deletion" rules mapping each article word to ϵ, include an additional deletion feature for these rules, and allow for an infinite distortion limit. We also explicitly tune the model using MERT to target the 75-byte capped ROUGE score as opposed to standard

Model	DUC-2004 ROUGE-1	DUC-2004 ROUGE-2	DUC-2004 ROUGE-L	Gigaword ROUGE-1	Gigaword ROUGE-2	Gigaword ROUGE-L	Ext. %
IR	11.06	1.67	9.67	16.91	5.55	15.58	29.2
PREFIX	22.43	6.49	19.65	23.14	8.25	21.73	100
COMPRESS	19.77	4.02	17.30	19.63	5.13	18.28	100
W&L	22	6	17	-	-	-	-
TOPIARY	25.12	6.46	20.12	-	-	-	-
MOSES+	26.50	8.13	22.85	28.77	12.10	26.44	70.5
ABS	26.55	7.06	22.05	30.88	12.22	27.77	85.4
ABS+	28.18	8.49	23.81	31.00	12.65	28.34	91.5
REFERENCE	29.21	8.38	24.46	-	-	-	45.6

Table 1: Experimental results on the main summary tasks on various ROUGE metrics . Baseline models are described in detail in Section 7.2. We report the percentage of tokens in the summary that also appear in the input for Gigaword as `Ext %`.

BLEU-based tuning. Unfortunately, one remaining issue is that it is non-trivial to modify the translation decoder to produce fixed-length outputs, so we tune the system to produce roughly the expected length.

7.3 Implementation

For training, we use mini-batch stochastic gradient descent to minimize negative log-likelihood. We use a learning rate of 0.05, and split the learning rate by half if validation log-likelihood does not improve for an epoch. Training is performed with shuffled mini-batches of size 64. The minibatches are grouped by input length. After each epoch, we renormalize the embedding tables (Hinton et al., 2012). Based on the validation set, we set hyperparameters as $D = 200$, $H = 400$, $C = 5$, $L = 3$, and $Q = 2$.

Our implementation uses the Torch numerical framework (`http://torch.ch/`) and will be openly available along with the data pipeline. Crucially, training is performed on GPUs and would be intractable or require approximations otherwise. Processing 1000 mini-batches with $D = 200$, $H = 400$ requires 160 seconds. Best validation accuracy is reached after 15 epochs through the data, which requires around 4 days of training.

Additionally, as described in Section 5 we apply a MERT tuning step after training using the DUC-2003 data. For this step we use Z-MERT (Zaidan, 2009). We refer to the main model as ABS and the tuned model as ABS+.

8 Results

Our main results are presented in Table 1. We run experiments both using the DUC-2004 evaluation data set (500 sentences, 4 references, 75 bytes) with all systems and a randomly held-out Gigaword test set (2000 sentences, 1 reference). We first note that the baselines COMPRESS and IR do relatively poorly on both datasets, indicating that neither just having article information or language model information alone is sufficient for the task. The PREFIX baseline actually performs surprisingly well on ROUGE-1 which makes sense given the earlier observed overlap between article and summary.

Both ABS and MOSES+ perform better than TOPIARY, particularly on ROUGE-2 and ROUGE-L in DUC. The full model ABS+ scores the best on these tasks, and is significantly better based on the default ROUGE confidence level than TOPIARY on all metrics, and MOSES+ on ROUGE-1 for DUC as well as ROUGE-1 and ROUGE-L for Gigaword. Note that the additional extractive features bias the system towards retaining more input words, which is useful for the underlying metric.

Next we consider ablations to the model and algorithm structure. Table 2 shows experiments for the model with various encoders. For these experiments we look at the perplexity of the system as a language model on validation data, which controls for the variable of inference and tuning. The NNLM language model with no encoder gives a gain over the standard n-gram language model. Including even the bag-of-words encoder reduces perplexity number to below 50. Both the convolutional encoder and the attention-based encoder further reduce the perplexity, with attention giving a value below 30.

We also consider model and decoding ablations on the main summary model, shown in Table 3. These experiments compare to the BoW encoding models, compare beam search and greedy decoding, as well as restricting the system to be com-

Model	Encoder	Perplexity
KN-Smoothed 5-Gram	none	183.2
Feed-Forward NNLM	none	145.9
Bag-of-Word	enc_1	43.6
Convolutional (TDNN)	enc_2	35.9
Attention-Based (ABS)	enc_3	27.1

Table 2: Perplexity results on the Gigaword validation set comparing various language models with C=5 and end-to-end summarization models. The encoders are defined in Section 3.

Decoder	Model	Cons.	R-1	R-2	R-L
Greedy	ABS+	Abs	26.67	6.72	21.70
Beam	BOW	Abs	22.15	4.60	18.23
Beam	ABS+	Ext	27.89	7.56	22.84
Beam	ABS+	Abs	28.48	8.91	23.97

Table 3: ROUGE scores on DUC-2003 development data for various versions of inference. Greedy and Beam are described in Section 4. Ext. is a purely extractive version of the system (Eq. 2)

plete extractive. Of these features, the biggest impact is from using a more powerful encoder (attention versus BoW), as well as using beam search to generate summaries. The abstractive nature of the system helps, but for ROUGE even using pure extractive generation is effective.

Finally we consider example summaries shown in Figure 4. Despite improving on the baseline scores, this model is far from human performance on this task. Generally the models are good at picking out key words from the input, such as names and places. However, both models will reorder words in syntactically incorrect ways, for instance in Sentence 7 both models have the wrong subject. ABS often uses more interesting re-wording, for instance *new nz pm after election* in Sentence 4, but this can also lead to attachment mistakes such a *russian oil giant chevron* in Sentence 11.

9 Conclusion

We have presented a neural attention-based model for abstractive summarization, based on recent developments in neural machine translation. We combine this probabilistic model with a generation algorithm which produces accurate abstractive summaries. As a next step we would like to further improve the grammaticality of the summaries in a data-driven way, as well as scale this system to generate paragraph-level summaries. Both pose additional challenges in terms of efficient alignment and consistency in generation.

I(1): a detained iranian-american academic accused of acting against national security has been released from a tehran prison after a hefty bail was posted , a to p judiciary official said tuesday .
G: iranian-american academic held in tehran released on bail
A: detained iranian-american academic released from jail after posting bail
A+: detained iranian-american academic released from prison after hefty bail

I(2): ministers from the european union and its mediterranean neighbors gathered here under heavy security on monday for an unprecedented conference on economic and political cooperation .
G: european mediterranean ministers gather for landmark conference by julie bradford
A: mediterranean neighbors gather for unprecedented conference on heavy security
A+: mediterranean neighbors gather under heavy security for unprecedented conference

I(3): the death toll from a school collapse in a haitian shanty-town rose to ## after rescue workers uncovered a classroom with ## dead students and their teacher , officials said saturday .
G: toll rises to ## in haiti school unk : official
A: death toll in haiti school accident rises to ##
A+: death toll in haiti school to ## dead students

I(4): australian foreign minister stephen smith sunday congratulated new zealand 's new prime minister-elect john key as he praised ousted leader helen clark as a " gutsy " and respected politician .
G: time caught up with nz 's gutsy clark says australian fm
A: australian foreign minister congratulates new nz pm after election
A+: australian foreign minister congratulates smith new zealand as leader

I(5): two drunken south african fans hurled racist abuse at the country 's rugby sevens coach after the team were eliminated from the weekend 's hong kong tournament , reports said tuesday .
G: rugby union : racist taunts mar hong kong sevens : report
A: south african fans hurl racist taunts at rugby sevens
A+: south african fans racist abuse at rugby sevens tournament

I(6): christian conservatives – kingmakers in the last two us presidential elections – may have less success in getting their pick elected in #### , political observers say .
G: christian conservatives power diminished ahead of #### vote
A: christian conservatives may have less success in #### election
A+: christian conservatives in the last two us presidential elections

I(7): the white house on thursday warned iran of possible new sanctions after the un nuclear watchdog reported that tehran had begun sensitive nuclear work at a key site in defiance of un resolutions .
G: us warns iran of step backward on nuclear issue
A: iran warns of possible new sanctions on nuclear work
A+: un nuclear watchdog warns iran of possible new sanctions

I(8): thousands of kashmiris chanting pro-pakistan slogans on sunday attended a rally to welcome back a hardline separatist leader who underwent cancer treatment in mumbai .
G: thousands attend rally for kashmir hardliner
A: thousands rally in support of hardline kashmiri separatist leader
A+: thousands of kashmiris rally to welcome back cancer treatment

I(9): an explosion in iraq 's restive northeastern province of diyala killed two us soldiers and wounded two more , the military reported monday .
G: two us soldiers killed in iraq blast december toll ###
A: # us two soldiers killed in restive northeast province
A+: explosion in restive northeastern province kills two us soldiers

I(10): russian world no. # nikolay davydenko became the fifth withdrawal through injury or illness at the sydney international wednesday , retiring from his second round match with a foot injury .
G: tennis : davydenko pulls out of sydney with injury
A: davydenko pulls out of sydney international with foot injury
A+: russian world no. # davydenko retires at sydney international

I(11): russia 's gas and oil giant gazprom and us oil major chevron have set up a joint venture based in resource-rich northwestern siberia , the interfax news agency reported thursday quoting gazprom officials .
G: gazprom chevron set up joint venture
A: russian oil giant chevron set up siberia joint venture
A+: russia 's gazprom set up joint venture in siberia

Figure 4: Example sentence summaries produced on Gigaword. **I** is the input, **A** is ABS, and **G** is the true headline.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Michele Banko, Vibhu O Mittal, and Michael J Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 318–325. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 137–144. Association for Computational Linguistics.

Hal Daumé III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 449–456. Association for Computational Linguistics.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*, pages 1–8. Association for Computational Linguistics.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *EMNLP*, pages 1481–1491.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hongyan Jing. 2002. Using hidden markov modeling to decompose human-written summaries. *Computational linguistics*, 28(4):527–543.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, pages 1700–1709.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81.

Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Generation with quasi-synchronous grammar. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 513–523. Association for Computational Linguistics.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. Association for Computational Linguistics.

Omar Zaidan. 2009. Z-mert: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.

David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*, pages 112–119.