Unsupervised Sentence Compression using Denoising Auto-Encoders

Thibault Fevry*
Center for Data Science
New York University
Thibault.Fevry@nyu.edu

Jason Phang*
Center for Data Science
New York University
jasonphang@nyu.edu

Abstract

In sentence compression, the task of shortening sentences while retaining the original meaning, models tend to be trained on large corpora containing pairs of verbose and compressed sentences. To remove the need for paired corpora, we emulate a summarization task and add noise to extend sentences and train a denoising auto-encoder to recover the original, constructing an end-to-end training regime without the need for any examples of compressed sentences. We conduct a human evaluation of our model on a standard text summarization dataset and show that it performs comparably to a supervised baseline based on grammatical correctness and retention of meaning. Despite being exposed to no target data, our unsupervised models learn to generate imperfect but reasonably readable sentence summaries. Although we underperform supervised models based on ROUGE scores, our models are competitive with a supervised baseline based on human evaluation for grammatical correctness and retention of meaning.

1 Introduction

Sentence compression is the task of condensing a longer sentence into a shorter one that still retains the meaning of the original. Past models for sentence compression have tended to rely heavily on strong linguistic priors such as syntactic rules or heuristics (Dorr et al., 2003; Cohn and Lapata, 2008). More recent work using deep learning involves models trained without strong linguistic priors, instead requiring large corpora consisting of pairs of longer and shorter sentences (Miao and Blunsom, 2016).

Sentence compression can also be can be seen as a "scaled down version of the text summarization problem" (Knight and Marcu, 2002). Within text summarization, two broad approaches exist: *extractive* approaches extract explicit tokens or phrases from the reference text, whereas *abstractive* approaches involve a compressed paraphrasing of the reference text, similar to the approach humans might take (Jing, 2000, 2002).

In the related domain of machine translation, a task that also involves learning a mapping from one string of tokens to another, state of the art models using deep learning techniques are trained on large parallel corpora. Recent promising work on unsupervised neural machine translation (Artetxe et al., 2017; Lample et al., 2017) has shown that with the right training regime, it is possible to train models for machine translation between two languages given only two unpaired monolingual corpora.

In this paper, we apply neural text summarization techniques to the task of sentence compression, focusing on on extractive summarization. However, we depart significantly from prior work by taking a fully unsupervised training approach. Beyond not using parallel corpora, we train our model using a single corpus. In contrast to unsupervised neural machine translation, which still uses two corpora, we do not have separate corpora of longer and shorter sentences.

We show that a simple denoising auto-encoder model, trained on removing and reordering words from a noised input sequence, can learn effective sentence compression, generating shorter sequences of reasonably grammatical text that retain the original meaning. While the models are still prone to both errors in grammar and meaning, we believe that this is a strong step toward reducing reliance on paired corpora.

We evaluate our model using both a stan-

* Denotes equal contribution

Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 413–422
Brussels, Belgium, October 31 - November 1, 2018.

dard text-summarization benchmark as well as human evaluation of compressed sentences based on grammatical correctness and retention of meaning. Although our models do not capture the written style of the target summaries (headlines), they still produce reasonably readable and accurate compressed sentence summaries, without ever being exposed to any target sentence summaries. We find that our model underperforms based on ROUGE metrics, especially compared to supervised models, but performs competitively with supervised baselines in human evaluation. We further show that providing the model with a sentence embedding of the original sentence leads to better ROUGE scores but worse human evaluation scores. However, both unsupervised and supervised methods still fall short based on human evaluation, and effective sentence compression and summarization remains an open problem.

2 Related work

Early sentence compression approaches were extractive, focusing on deletion of uninformative words from sentences through learned rules (Knight and Marcu, 2002) or linguistically-motivated heuristics (Dorr et al., 2003). The first abstractive approaches also relied on learned syntactic transformations (Cohn and Lapata, 2008).

Recent work in automated text summarization has seen the application of sequence-to-sequence models to automatic summarization, including both extractive (Nallapati et al., 2017) and abstractive (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; Paulus et al., 2017; Fan et al., 2017) approaches, as well as hybrids of both (See et al., 2017). Although these methods have achieved state-of-the-art results, they are constrained by their need for large amounts paired document-summary data.

Miao and Blunsom (2016) seek to overcome this shortcoming by training separate compressor and reconstruction models, allowing for training based on both paired (supervised) and unlabeled (unsupervised) data. For their compressor, they train a discrete variational auto-encoder for sentence compression and use the REINFORCE algorithm to allow end-to-end training. They further use a pre-trained language model as a prior for their compression model to induce their compressed output to be grammatical. However, their reported results are still based on models trained on at least 500k instances of paired data.

In machine translation, unsupervised methods for aligning word embeddings using only unmatched bilingual corpora, trained with only small seed dictionaries, (Mikolov et al., 2013; Lazaridou et al., 2015), adversarial training on similar corpora (Zhang et al., 2017; Conneau et al., 2017b) or even on distant corpora and languages (Artetxe et al., 2018) have enabled the development of unsupervised machine translation (Artetxe et al., 2017; Lample et al., 2017). However, it is not clear how to adapt these methods for summarization where the task is to shorten the reference rather than translate it. Wang and Lee (2018) train a generative adversarial network to encode references into a latent space and decode them in summaries using only unmatched document-summary pairs. However, in contrast with machine translation where monolingual data is plentiful and paired data scarce, summaries are paired with their respective documents when they exist, thus limiting the usefulness of such approaches. In contrast, our method requires no summary corpora.

Denoising auto-encoders (Vincent et al., 2008) have been successfully used in natural language processing for building sentence embeddings (Hill et al., 2016), training unsupervised translation models (Artetxe et al., 2017) or for natural language generation in narrow domains (Freitag and Roy, 2018). In all those instances, the added noise takes the form of random deletion of words and word swapping or shuffling. Although our noising mechanism relies on adding rather than removing words, we take some inspiration from these works.

Work in sentence simplification (see Shardlow (2014) for a survey) has some similarities with sentence compression, but it differs in that the key focus is on making sentences more easily understandable rather than shorter. Though word deletion is used, sentence simplification methods feature sentence splitting and word simplification which are not usually present in sentence compression. Furthermore, these methods often rely heavily on learned rules (e.g lexical simplification as in Biran et al. (2011)), integer linear programming and sentence parse trees which makes them starkly different from our deep learning-based approach. The exceptions that adopt end-to-end approaches, such as Filippova et al. (2015), are usually supervised and focus on word deletion.

3 Methods

3.1 Model

Our core model is based on a standard attentional encoder-decoder (Bahdanau et al., 2014), consisting of multiple layers bi-directional long short-term memory networks in both the encoder and decoder, with negative-log likelihood as our loss function. We detail below the training regime and model modifications to apply the denoising auto-encoding paradigm to sentence compression.

3.2 Additive Noising

Since we do not use paired sentence compression data with which to train our model in a supervised way, we simulate a supervised training regime by modifying a denoising auto-encoder (DAE) training regime to more closely resemble supervised sentence compression. Given a reference sentence, we extend and shuffle the input sentence, and then train our model to recover the original reference sentence. In doing so, the model has to exclude and reorder words, and hence learns to output shorter but grammatically correct sentences.

Additive Sampling We randomly sample additional sentences from our data set, and then subsample a number of words from each without replacement. We then append the newly sampled words to our reference sentence. In our experiments, we sample two additional sentences for each reference sentence, and the number of words sampled from each is dependent on the length of the original reference sentence. In practice, we aim to generate a noised sentence that extends the original sentence by 40% to 60%. To fit the fully unsupervised learning paradigm, we do not introduce any biases into our sampling of words in training our model. In particular, we excluded approaches that overweighted adjectives or speaker identification (e.g "said X on Tuesday") in noising.

Shuffling Next, we shuffle the resultant string of words. We experiment with two forms of shuffling: (i) a complete word (unigram) shuffle and (ii) bigram shuffling, where we only shuffle among the word bigrams, keeping pairs of adjacent words together.

This process is illustrated in Figure 1.

3.3 Length Countdown

To induce our model to output sequences of a desired length, we augment the RNN decoder in our model to take an additional *length countdown* input. In the context of text generation, RNN decoders can be formulated as follows:

$$h_t = \text{RNN}(h_{t-1}, x_t) \tag{1}$$

where h_{t-1} is the hidden state at the previous step and x_t is an external input (often an embedding of the previously decoded token). Let T_{dec} be the desired length of our output sequence. We modify (1) with an additional input:

$$h_t = \text{RNN}(h_{t-1}, x_t, T_{\text{dec}} - t) \tag{2}$$

The length countdown $T - t$ is a single scalar input that ticks down to 0 when the decoder reaches the desired length T, and goes negative after. In practice, $(x_t, T_{\text{dec}} - t)$ are concatenated into a single vector. We also experimented with adding a length penalty to our objective function to amplify the loss from predicting the end-of-sequence token <EOS> at the desired time step, but did not find that our models required this additional loss term to output sequences of the desired length.

Explicit length control has been used in previous summarization work. Fan et al. (2017) introduced a length marker token that induces the model to target an output of a desired length, coarsely divided into discrete bins. Kikuchi et al. (2016) examined several methods of introducing target output length information, and found that they were effective without negatively impacting summarization quality. We found more success with our models with a per time-step input compared to a token at the start of the sequence as in Fan et al. (2017).

3.4 Input Sentence Embedding

The model specified above is supplied only with an unordered set of words with which to construct a shorter sentence. However, there are typically many ways of ordering a given set of words into a grammatical sentence. In order to help our model better recover the original sentence, we also provide the model with an InferSent sentence embedding (Conneau et al., 2017a) of the original sentence, generated using a pre-trained InferSent model. The InferSent model is trained on NLI tasks, where, given a longer premise text and a

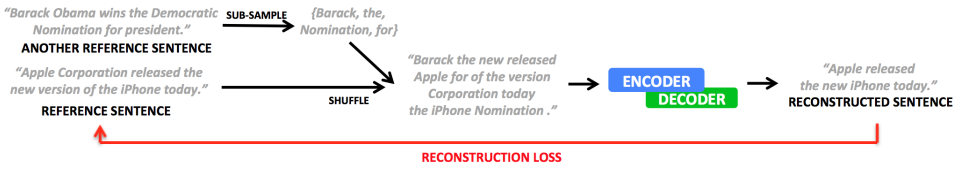

Figure 1: Illustration of Additive Noising. A reference sentence is noised with subsampled words from another sentence, and then shuffled. The denoising auto-encoder is trained to recover the original reference sentence. This simulates a text summarization training regimes without the need for parallel corpora.

shorter hypothesis text, the model is required to determine if the premise (i) entails, (ii) contradicts or (iii) is neutral to the hypothesis. The InferSent sentence embeddings are an intermediate output of the model, reflecting information captured from each text string. Conneau et al. show that InferSent sentence embeddings capture various aspects of the semantics of a string of text (Conneau et al., 2018), and should provide additional information to the model as to which ordering of words best match the meaning original sentence.

We incorporate the InferSent embeddings by modifying the hidden state passed between the encoder and the decoder. In typical RNN encoder-decoder architectures, the final hidden state of the encoder is used as the initial hidden state of the decoder. In other words, $h_0^{\text{dec}} = h_{T_{\text{enc}}}^{\text{enc}}$. We learn a fully connected layer f to be used as follows:

$$h_0^{\text{dec}} = f(h_{T_{\text{enc}}}^{\text{enc}}, s) \quad (3)$$

where s is the InferSent embedding of the input sentence. This transformation is only applied once on the hidden state shared from the encoder to the decoder. In the case of LSTMs, where there are both hidden states and cell states, we learn a fully connected mapping for each.

3.5 Numbered Out-of-Vocabulary (OOV) Embeddings

Many text summarization data sets are based on news articles and headlines, which often include names, proper nouns, and other rare words or tokens that may not appear in word embedding dictionaries. In addition, the output layer of most models are based on a softmax over all potential output tokens. This means that expanding the vocabulary to potentially include more rare words increases computation and memory costs in the final layer linearly. There are many approaches to tackle out-of-vocabulary (OOV) tokens (See et al., 2017; Nallapati et al., 2016), and we detail below our approach.

To address the frequent occurrences of OOV characters, we learn a fixed number of embeddings for numbered OOV tokens.[2] Given an input sequence, we first parse the sentence to identify OOV tokens and number them in order, while storing the map from numbered OOV tokens to words.[3] When embedding the respective tokens to be inputs to the RNN, we assign the corresponding embeddings for each numbered OOV token. We apply the same numbering system to the target, so the same word in the input and output will always be assigned the same numbered OOV token, and hence the same embedding. At inference, we replace any output numbered OOV tokens with their respective words. This allows us to output sentences using words not in our vocabulary.

This approach is similar to the pointer-generator model (See et al., 2017), but whereas See et al. compute attention weights over all tokens in the input to learn where to copy and have an explicit switch between copying (pointer) and output (generator), we learn embeddings for a fixed number of OOV tokens, and the embeddings are in the same latent space as our pre-trained word embeddings.

4 Experimental Setup

4.1 Data

For our text summarization task, We use the Annotated Gigaword (Napoles et al., 2012) in line with Rush et al. (2015). This data set is derived for news articles, and consists of pairs of the main sentences in the article (longer), and the headline (shorter). The former and latter are used as references and summaries respectively in the context of summarization tasks. We preprocess the data using the scripts made available by the authors, which produces about 3.8M training examples and 400K validation examples. We sample randomly

[2] We use a fixed number of 10 OOV tokens in our experiments.

[3] In the case of shuffling and noising, we number the OOV tokens before shuffling, and number any additional OOV tokens from the noised input sentence in a second pass.

10K examples for validation and 10K for testing from the validation set, similar to the procedure in Nallapati et al. (2016). Like Rush et al. (2015), we only extract the tokenized words of the first sentence, in contrast with Nallapati et al. (2016) who extract the first two sentences as well as part-of-speech and named-entities tags.

4.2 Training

In training, we only use the reference sentences from the Gigaword dataset. For all our models, we used GloVe word embeddings (Pennington et al., 2014). We freeze these embeddings during training. Our vocabulary is comprised of the 20000 most frequent words in the references, and we use the aforementioned numbered OOV embeddings for other unseen words. We similarly freeze the InferSent model for sentence embeddings. The encoder and decoder are both 3-layer LSTMs with 512 hidden units. We use a batch size of 128, and optimize our models using Adam (Kingma and Ba, 2014) with a initial learning rate of 0.0005, annealing it by 0.9 at every 10K mini-batches. We do not use dropout but use gradient clipping at 2. We train our models for 4 full epochs.

4.3 Inference

At inference, we supply our model with the unmodified reference sentences–hence no noising is applied. We use the length countdown to target outputs of half the length of the reference sentences. The application of sentence embeddings is unchanged from training.

4.4 Implementation

We implemented our models using Pytorch (Paszke et al., 2017), and will make our code publicly available at `https://github.com/zphang/usc_dae`.

5 Results

5.1 ROUGE Evaluation

In Table 1, we evaluate our models on ROUGE (Lin, 2004) F1 scores, where a higher score is better. We provide a comparison with a simple but strong baseline, *F8W* is simply first 8 words of the input, as is done in Wang and Lee (2018) and similarly to the *Prefix* baseline (first 75 bytes) of Rush et al. (2015), as well as the ROUGE of the whole text with the target. We provide scores of two supervised text-summarization methods on Gigaword. One is our own baseline, consisting of a sequence-to-sequence attentional encoder-decoder trained on pairs of reference and target summary text, but incorporating the same length countdown mechanism as in our unsupervised models. The other is the *words-lvt2k-1sent* model of Nallapati et al. (2016). Although not their best model, it is most comparable to ours since it only uses the first sentence and does not extract *tf-idf* vectors nor named entities tags.

F8W and *All text* are strong baselines due to the tendency of news articles to contain specific terms that are rarely rephrased. We find that our models perform competitively with these baselines, although they pale in comparison to supervised methods, likely because they do not learn any style transfer and use only the reference's vocabulary and writing style. While our ROUGE-1 scores are in line with the baselines, our ROUGE-2 scores fall somewhat behind. Including InferSent sentence embeddings improves our ROUGE scores across the board. Our supervised baseline performance is close to that of Nallapati et al. (2016), with results lower in ROUGE-2 likely due to their use of beam search. Nevertheless, the supervised baseline is representative of the performance of a standard sequence-to-sequence attentional model on this task.

A direct comparison of ROUGE scores is not completely adequate for evaluating our model. Because of our training regime, our model primarily learned to generate shortened sentences that often still retain the style of the input sentences. Unlike other model setups, our model has never been exposed to any examples of summaries, and hence never adapts its output to match the style of the target summaries. In the case of Gigaword, the summaries are headlines from news articles, which are written in a particular linguistic style (e.g. dropping articles, having clauses rather than full sentences). ROUGE will thus penalize our model, that tends to output longer, full sentences. In addition, ROUGE is an imperfect metric for summarization as word/n-gram overlap does not fully capture summary relevancy and retention of meaning.[4] For this reason, we also conduct a separate human evaluation of our different models against

[4] See discussion in Nallapati et al. (2016), or in Paulus et al. (2017) where a reinforcement learning model trained on a Rouge-L objective alone achieves the best scores but "produces the least readable summaries among [their] experiments"

Example 1:
I: nearly ### of the released hostages remain in hospital , and more than ### of them are in very serious condition , russian medical authorities said sunday .
G: nearly ### people still hospitalized more than ### in critical condition
2-g shuf: more than ### hostages are in serious condition , russian medical authorities said .
2-g shuf + InferSent: nearly ### hostages of the nearly released in serious medical condition , said .

Example 2:
I: french president jacques chirac arrived here friday at the start of a <unk> during which he is expected to hold talks with romanian leaders on bucharest 's application to join nato .
G: chirac arrives in romania
2-g shuf: french president jacques chirac arrived here friday to hold talks with romanian leaders on nato .
2-g shuf + InferSent: french president jacques chirac arrived here friday at the start of talks to join nato .

Example 3:
I: swedish truck maker ab volvo on tuesday reported its third consecutive quarterly loss as sales plunged by one-third amid weak demand in the april-june period .
G: volvo posts $ ### million loss on falling sales
2-g shuf: swedish truck maker volvo ab on tuesday reported its third consecutive quarterly .
2-g shuf + InferSent: swedish truck maker ab volvo on tuesday reported its consecutive quarterly loss .

Example 4:
I: wall street stocks rallied friday as a weak report on us economic growth boosted hopes for an easier interest rate policy from the federal reserve and investors reacted to upbeat earnings news .
G: wall street shrugs off weak gdp pushes higher
2-g shuf: wall street stocks rallied friday as investors reacted to upbeat economic news and interest rate .
2-g shuf + InferSent: wall street stocks rallied friday as investors reacted to an economic growth report on hopes .

Figure 2: Examples of inputs, ground-truth summaries, and outputs from two of our models. **I** is input, **G** (gold) is the true summaries. Example 1 and 2 show our models summarizing pertinent information from the input. Example 3 demonstrates the ability to recover long ordered strings of tokens, even though the models are trained on shuffle data. Example 4 shows cases where the models output grammatical but semantically incorrect sentences.

a supervised baseline (Section 5.4).

5.2 ROUGE Ablation study

In Table 2, we report the results of an ablation study. We observe that all three components we vary, namely the use of attention, bigram shuffling, and incorporation of sentence embeddings, contribute positively to the performance of our model as measured by ROUGE. The model that incorporates all three obtains the highest ROUGE scores.

5.3 Impact of Length

To assess our models' ability to deal with sequences of text of different length, we measure the ROUGE scores on two bins of length of the input text, from 16 to 30 tokens and from 31 to 45. As expected, longer sentences pose a harder challenge to the model, with our model performing better on shorter than longer sentences. Across most sequence-based problems, models tend to perform better on shorter sequences. However, in the context of the text summarization or sentence compression, longer sentences not only contain more information that the model would need to selective remove, but also more information from which to identify the central theme of the sentence.

5.4 Human Evaluation

To qualitatively evaluate our model, we take inspiration from the methodology of Turner and Charniak (2005) to design our human evaluation. We asked 6 native English speakers to evaluate randomly chosen summaries from five models: our best models with and without InferSent sentence embeddings, a summary generated from a trained supervised model, and the ground truth summary. The sentences are evaluated based on two separate criteria: the grammaticality of the summary and how well it retained the information of the original sentence. In the former, only the summary is provided, whereas in the latter, the evaluator is shown both the original sentence as well as the summary. Each of these criteria were graded on a scale from 1 to 5. The examples are from the test set, with 50 examples randomly sampled for each evaluator and criterion.[5]

We report the average evaluation given by our 6 evaluators in Table 4. That the *Meaning* score for the ground truth is somewhat low (3.87) is not surprising. Within the Gigaword dataset, summaries (headlines) sometimes include information not within the reference (main line of the article). We observe that quantitative evaluation does not correlate well with human evaluation. Methods using InferSent embeddings improved our ROUGE scores but perform worse in human evaluation, which is in line with the summaries presented in 2. Notably, the model trained on shuffled bigrams and InferSent embeddings performed best within our ablation study, but the worst among the three models in human evaluation. Encouragingly, the model without InferSent embeddings performs competitively with the supervised baseline in both grammar and meaning scores, indicating that although it does not capture the style of headlines, it succeeds in generating grammatical sentences that

[5]The sampling is constrained to ensure each evaluator sees an equal number of summaries from each model, although evaluators are informed neither about the sampling process, nor how many or what models are involved.

Model	ROUGE			Avg. Length
	R-1	R-2	R-L	
Baselines:				
All text	**28.91**	**10.22**	**25.08**	31.3
F8W	26.90	9.65	25.19	8
Unsupervised (Ours):				
2-g shuf	27.72	7.55	23.43	15.4
2-g shuf + InferSent	**28.42**	**7.82**	**24.95**	15.6
Supervised abstractive:				
Seq2seq	**35.50**	15.54	32.45	15.4
(words-lvt2k-1sent) (Nallapati et al., 2016)	34.97	**17.17**	**32.70**	-

Table 1: Performance of Baseline, Unsupervised and Supervised Models. Our unsupervised models pale in comparison to supervised models, and perform in line with baselines. Simple baselines in text summarization benchmarks tend to be unusually strong. The unsupervised model incorporating sentence embeddings performs slightly better on ROUGE.

Model	ROUGE		
	R-1	R-2	R-L
1-g shuf (w/o attn)	23.01	5.51	20.07
2-g shuf (w/o attn)	22.36	5.18	19.60
1-g shuf	27.22	7.63	23.55
2-g shuf	27.72	7.55	23.43
1-g shuf + InferSent	28.12	7.75	24.81
2-g shuf + InferSent	**28.42**	**7.82**	**24.95**

Table 2: Ablation study. We find that using attention, shuffling bigrams, and incorporating sentence embeddings all improve our ROUGE scores. All length countdowns settings are the same is in the main model.

Input Length	ROUGE			Avg. Length
	R-1	R-2	R-L	
16-30	30.79	9.20	27.73	12.6
31-45	26.89	6.76	23.04	17.7

Table 3: Effect of input sentence length on performance, using the 2-g shuf + InferSent model. Performance tends to be worse on longer input texts.

Model	Grammar	Meaning
2-g shuf	**3.53 (±0.18)**	2.53 (±0.16)
1-g shuf + InferSent	2.82 (±0.17)	2.50 (±0.15)
2-g shuf + InferSent	2.87 (±0.16)	2.13 (±0.13)
Seq2seq (Supervised)	3.43 (±0.18)	**2.60 (±0.17)**
Ground Truth	4.07 (±0.13)	3.87 (±0.16)

Table 4: Human Evaluation. Mean scores, with 1 standard error confidence bands in parentheses. Our best model performs competitively with a supervised baseline in both grammatical correctness and retuention of meaning. Models with sentence embeddings perform worse in human evaluation, despite obtaining better ROUGE scores.

roughly match the meaning in the reference. Some evaluators highlighted that it was problematic to rate meaning for ungrammatical sentences.

5.5 Output Analysis

We show in Figure 2 several examples of the inputs, ground-truths target summaries, and outputs from 2 of our models. We observe that the output sentences are generally well-conditioned though occasionally imperfectly grammatical. We also observe certain artifacts from training only on reference texts that are not reflected in ground-truth summaries. For example, every output sentence ends with a period, and several examples end with speaker identification clauses. In all instances, we observe that the model without InferSent outputs sentences are more readable and relevant, confirming human evaluation results in 4.

Example 1 shows that our model can extract the most pertinent information to generate a grammatical summary that captures the original meaning.

Example 2 shows an instance where our output accurately summarizes the input text despite low

ROUGE scores to the target (R-1 of 21.1 and R-2 of 11.8). In this case, both models capture the core meaning of the input.

Example 3 shows that although the models are provided completely shuffled words in training, at inference it is able to recover complex terms such as "swedish truck maker ab volvo". We note that this may be a bias in the data set (sentences in news often start with proper nouns preceded by qualifiers) and hence a simple strategy for the model to discover. This examples also shows common mistake of our models: in the output of model without InferSent, it drops an important word ("loss") right before the end of the sentence, causing it to fail to capture the original meaning.

Example 4 shows that on longer sentences, our models may sometimes fail to accurately capture meaning. In this case, for the model without InferSent, although the output is grammatical and meaningful, it captures a meaning different than that of the original input. Indeed, our model suggests that upbeat news cause the rally whereas the original sentence indicates that given poor economic news investors anticipate easier monetary policy and thus caused a stock rally.

5.6 Length Variation

Because the desired length of the output sequence is a user-defined input in the model, we can take an arbitrary sentence and use the model to output the corresponding compressed (or even expanded) sentence of any desired length. We show two examples in Figure 3, where we vary the desired length from 7 to the input length, using our best model based on human evaluation. We observe that for very short desired lengths, the model struggles to produce meaningful sentences, whereas for desired lengths close to the input length, the model nearly reconstructs the input sentence. Nevertheless, we observe that for many of the intermediate lengths, the model outputs sentences that are close in meaning to the input sentence, with different ways of rephrasing or shortening the input sentence in the interim. This suggests that when the ratio of the desired output sentence length to the input sentence length is close to that of the training regime, the model is able to perform better than when it has to generate sentences with other ratios.

Example 1:
I: three convicted serial killers have been hanged in tehran 's evin prison , the khorasan newspaper reported sunday .
L=9: three convicted serial killers have been hanged in .
L=11: three convicted serial killers have been hanged in prison sunday .
L=13: three convicted serial killers have been hanged , a newspaper reported sunday .
L=15: three convicted serial killers have been hanged in tehran , a newspaper reported sunday .
L=17: three convicted serial killers have been hanged in tehran , the tehran 's newspaper reported sunday .
L=19: three convicted serial killers have been hanged in tehran 's prison , the newspaper tehran newspaper reported sunday .

Example 2:
I: a home-made bomb was found near a shopping center on indonesia 's ambon island , where ## people were wounded by an explosion at the weekend , state media said on monday .
L=9: a home-made bomb explosion wounded ## people monday .
L=11: a home-made bomb explosion wounded ## people on indonesia monday .
L=13: a home-made bomb explosion wounded ## people on indonesia 's ambon island .
L=15: a home-made bomb explosion wounded ## people at a shopping center on ambon monday .
L=17: a home-made bomb explosion wounded ## people at a shopping center on ambon island on monday .
L=19: a home-made bomb explosion wounded ## people at a shopping center near ambon on indonesia 's island state .
L=21: a home-made bomb explosion wounded ## people at a shopping center near ambon on indonesia 's ambon island on monday .
L=23: a home-made bomb was found on a shopping center near ambon , indonesia 's state on monday , state media said monday .
L=25: a home-made bomb was found on a shopping center near ambon , indonesia 's state media center where ## people were wounded by bomb .
L=27: a home-made bomb was found on a shopping center near ambon , indonesia 's state media center where ## people were wounded , media said monday .
L=29: a home-made bomb was found on a shopping center near ambon , indonesia 's state media on monday , where ## people were wounded by an explosion nearby .
L=31: a home-made bomb was found on a shopping center near ambon , indonesia 's state media on monday , where ## people were wounded by an explosion at the weekend .
L=33: a home-made bomb was found on a shopping center near ambon , indonesia 's state media on monday , where ## people were wounded by an explosion at the weekend on monday .

Figure 3: Summaries of varied desired lengths, using the 2-g shuf model. **L** is the desired output length provided to the model. Because the desired output length is a human-provided input, we can produce summaries of varying lengths, ranging from highly contracted to verbose.

6 Discussion

In our experiments, we found that denoising auto-encoders quickly learn to generate well-conditioned text, even from badly conditioned inputs. We were surprised by the ability of denoising auto-encoders to recover readable sentences even from completely shuffled and noised sets of words. We observed some cases where the denoising auto-encoders outputs sequences that are grammatical correct but nonsensical or semantically different from the input. However, the ability for denoising auto-encoders to subsample words to form grammatical sentences would significantly reduce the search space for candidate sentences,

and we believe this could be useful for tasks involving sentence construction and reformulation.

Our attempts to better condition the denoising auto-encoders outputs on the original sentence using sentence embeddings had mixed results. Although the incorporation of InferSent embeddings improved our quantitative ROUGE scores, human evaluators scored outputs conditioned on InferSent embeddings markedly worse on both grammar and meaning retention. It is unclear whether this is due to InferSent embeddings failing to capture the most significant semantic information, or if our mechanism for incorporating the sentence embedding is suboptimal.

Lastly, we echo sentiments from previous authors that ROUGE remains an imperfect proxy for measuring the adequacy of summaries. We found that ROUGE scores can be fairly uncorrelated with human evaluation, and in general can be distorted by quirks of the data set or model outputs, particularly pertaining to length, formatting, and handling of special tokens. On the other hand, human evaluation can be more sensitive to comprehensibility and relevancy while being more robust to rewording and reasonable ambiguity. Based on our human evaluation, we find that both unsupervised and supervised methods still fall short of effective sentence compression and summarization.

7 Conclusion

We present a fully unsupervised approach to the task of sentence compression in the form of a denoising auto-encoder with additive noising and word shuffling. Our model achieves comparable scores in human evaluation to a supervised sequence-to-sequence attentional baseline in grammatical correctness and retention of meaning, but underperforms on ROUGE. Output analysis indicates that our model does not capture the particular style of the summaries in the Gigaword dataset, but nevertheless produces reasonably valid sentences that capture the meaning of the input. Although our models are still prone to making mistakes, they provide a strong baseline for future sentence compression and summarization work.

Acknowledgments

We would like to express our deepest gratitude to Sam Bowman for his thoughtful advice and feedback in the writing of this paper. We thank the NVIDIA Corporation for their support.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv preprint 1805.06297*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint 1710.11041*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of ACL*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of NAACL*.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of ACL*, pages 137–144.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017a. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loic Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv e-prints*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017b. Word translation without parallel data. *arXiv preprint 1710.04087*.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of NAACL*, pages 1–8.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *arXiv preprint 1711.05217*.

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

Markus Freitag and Scott Roy. 2018. Unsupervised natural language generation with denoising autoencoders. *arXiv preprint arXiv:1804.07899*.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint 1602.03483*.

Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *Proceedings of ANLP*, pages 310–315.

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

Y. Kikuchi, G. Neubig, R. Sasano, H. Takamura, and M. Okumura. 2016. Controlling Output Length in Neural Encoder-Decoders. *ArXiv e-prints*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint 1412.6980*.

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

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint 1711.00043*.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of ACL*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL*, page 10.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *EMNLP*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint 1309.4168*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint 1602.06023*.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of AKBC-WEKEX*, pages 95–100.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint 1705.04304*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

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

Abigail See, Peter Liu, and Christopher Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint 1704.04368*.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of ACL*, pages 290–297.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103.

Yau-Shian Wang and Hung-Yi Lee. 2018. Learning to encode text as human-readable summaries using generative adversarial networks.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of ACL*.