Experiments in Unsupervised Text Summarization

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

Deep learning methods for text summarization have generally required supervised training on paired text and summary data. Following recent work on unsupervised neural machine translation, we propose a series of techniques to train a model capable of summarizing text without the need for parallel corpora. Building on a standard attentional encoder-decoder model, we induce the model to generate shorter sequences of text while conditioning the generated text to retain the meaning of the input. We evaluate the efficacy of two main methods: back-expansion / summarization and denoising auto-encoders. Although both models pale in comparison to strong summarization baselines on ROUGE scores, we find that the latter is capable of producing highly readable summaries that capture the meaning of the reference text.

1 Introduction

As in the case of Neural Machine Translation, training neural network models for automatic text summarization is constrained by the availability of parallel corpora – corpora consisting of pairs of longer reference texts and their corresponding summaries. Taking inspiration from unsupervised machine translation (Artetxe et al., 2017; Lample et al., 2017), we propose two approaches to neural text summarization without parallel corpora: back-expansion / summarization (BE/S), wherein we train a model construct a summary/expansion of the reference text, and then reconstruct the original input, and denoising auto-encoders. Our goal is not to beat existing state-of-the-art methods, but to evaluate the plausibility unsupervised approaches to text summarization.

2 Related work

The goal of summarization is to produce a condensed version of an input text that is semantically close to the original. Hereafter, we refer to the longer text as the 'reference text' and the shorter output as the 'summary'. Two broad approaches exist: *extractive summarization* extracts explicit tokens or phrases from the reference text whereas *abstractive summarization* involves a compressed paraphrasing of the reference text, similar to the approach humans might take (Hongyan, 2002).

Most early summarization work is extractive and seeks keep the most informative words, or alternatively to delete uninformative words from the reference text. Knight and Marcu (2002) first produces the parse tree of the reference text, and then learns a series of rules for reducing the size of that tree. Early abstractive work leverages linguistically-motivated heuristics (Dorr et al., 2003) or learned syntactic transformations (Cohn and Lapata, 2008) to transform the input to create a suitable summary.

Recent approaches to summarization are supervised, largely using encoder-decoder setups for both for extractive (Nallapati et al., 2017) and abstractive summarization (Rush et al., 2015; Chopra et al., 2016; See et al., 2017). Approaches such as See et al.'s can be seen as hybrids of both, with both an abstractive attentional encoder-decoder, and a more extractive Pointer network to extract words directly from the reference text. While incorporating extractive strategies has been shown to improve summaries based on ROUGE metrics, they also introduce greater model complexity. In this paper, we leaned away from more complex models and opted for the basic attentional encoder-decoder.

Our work is motivated by the early success of unsupervised neural machine translation methods

(Artetxe et al., 2017; Lample et al., 2017). Although models trained without parallel corpora under-perform supervised models, they demonstrate the ability of deep learning models to learn informative mappings between two languages without paired training data. The approach of *backtranslation*, where a model is fed its own translations from one langauge to another and tasked with translating it back is the inspiration for our *back-expansion* approach. These works also make use of data-augmentation and "noising" of the input text in training their models, and we build on these in our denoising auto-encoder models. The key difference between the two unsupervised tasks is that whereas the unsupervised translation models have access to both corpora, albeit unpaired, our unsupervised summarization model never sees any summaries, and hence needs to learn linguistic compression without ever being shown how summaries are written.

Aside from unsupervised translation, denoising auto-encoders introduced by (Vincent et al., 2008) have been used in NLP for building sentence representations (Hill et al., 2016) (word deletion + and swapping neighbors) and language generation (Freitag and Roy, 2018) (shuffling + word deletion). In the case of latter, common/stop-words in the sentence are dropped and the remaining words are shuffled to form a set of "keywords". The model is then trained to reconstruct the original sentence. We take some inspiration from this work in the additive-sampling in our denoising auto-encoder.

3 Methods

We detail below our proposed strategies for the unsupervised training of a text-summarization model. We assume that the reference and summary domains share the same vocabulary.

3.1 Model

Our summarization model follows the standard attentional encoder-decoder setup (Bahdanau et al., 2014). Our primary modification to the model is the the incorporation of the *length countdown* as input, detailed below.

3.2 Length Countdown

To induce our model to generate shorter sequences of text than the reference, we augment our decoder to accept an additional input that counts down to the desired output length. The desired length is user-specified, although in practice it can be a function of the input length. At each decoding time-step, the decoder accepts an additional integer of the time-steps remaining before it should stop decoding. At the final time-step, it is supplied with the number 0, and negative integers after. Figure 1 illustrates this mechanism. To train the model to abide by the desired output length, we add a **length penalty** that is computed based on the log-loss from predicting the end-sequence token <EOS> relative to other tokens at the desired time-steps, and the converse at all other time-steps.

Having the length be a user-defined input provides several benefits. Firstly, we can induce the model to generate sequences shorter as well as longer than the input sequence - this is important for the back-expansion method below. Furthermore, the initial training of the model is more stable given the fixed desired length, compared to having the model determine its own optimal output length, which induces many edge-cases (e.g. empty outputs).

3.3 Back-Expansion / Summarization (BE/S)

Given a good summary, we ought to be able to recover with some degree of accuracy the original input. This is the idea behind **back-expansion**, where we have the model first *summarize* an input, and then have it *expand* the summary back into the original string. This turns our unsupervised learning setup into a supervised one, as the ground-truth labels from the resultant expansion is the reference text. Figure 2 illustrates how this works in practice. We use the same model for summarization and expansion. To allow our model to take in the outputs of the decoder, we need to discretize the outputs of the decoder while allowing gradients to flow through. Here, we use the Gumbel-Softmax reparameterization trick (Jang et al., 2016) as applied in (Gu et al., 2017) with optional temperature annealing. Since the model is now fully differentiable from the reconstruction to the original string, it can be trained end-to-end.

To maintain symmetry in our training regime, we also perform **back-summarization**, where we first expand the text, then summarize it to the original. In both cases, the final goal is to reproduce the original string of text after two passes through our model, and in training, the desired length of the intermediate output is randomly sampled within a

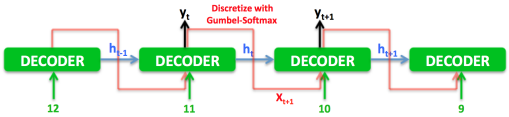

Figure 1: Illustration of Length Countdown

Figure 2: Back-Expansion

user-defined range.

One issue with back-summarization is that it imposes no constraints on the intermediate output sequence. Our early experiments showed that without further conditioning on the intermediate output, the generated sequences can become unintelligible while still allowing the model to reproduce the original sequence in the second phase. Hence, we specify below additional constraints to better condition our intermediate outputs.

3.4 Semantic Loss

To ensure that the generated summary is semantically consistent with the reference text, we incorporate semantic-based losses in our training. We leverage the InferSent model (Conneau et al., 2017), which is based on NLI tasks, to derive our semantic-based loss.

An NLI task consists of an agent or model being given a longer premise text and a shorter hypothesis text, and being asked to choose whether the premise *entails* the hypothesis, the hypothesis *contradicts* the premise, or neither (*neutral*). The output of NLI models is thus a 3-way categorical classification. Under this paradigm, we expect our reference text to *entail* our generated summary. Hence, using a pre-trained InferSent model, we feed the input string into the InferSent model as well as the summaries from our model and compute a penalty based on the log-loss of InferSent not choosing *Entailment*.

3.5 Auto-Encoding

One approach to inducing the intermediate output to be intelligible is to also train the model to auto-encode - in other words, generating the original sequence directly, where the desired length is the input length. We found that introducing auto-encoding into our training quickly induced the model to output readable sequences, but with the downside of teaching the model to frequently copy text. We randomly alternate between summarization and auto-encoding with each training iteration.

3.6 Denoising Auto-Encoding with Additive Sampling

To prevent the model from only learning to copy from the reference text, we introduced noise into our input text via an "additive sampling" strategy. Given a reference sentence, we sample one or more additional reference sentences and subsample words from it to add to the reference sentence. We then shuffle the result and feed it into the encoder-decoder, and use it to reconstruct the original sentence. This process is illustrated in Figure 3. This means that the model will need to (i) identify the topic of the sentence from the set of given words, (ii) filter out the unrelated words (i.e. the added words), and (iii) reconstruct the original sentence by reordering the words. By adding 'irrelevant' words to the original sentence, the goal is to simulate the task of summarization, which we could re-frame as consisting of a reference sentence that contains a 'significant' part as well as 'less relevant' details that the model would omit. At inference, the model is simply provided with the full ordered reference text.

Although this task appears fairly ambitious given how little information the model is provided with, our early experiments show that a simple encoder-decoder is surprisingly capable at this task, at least for the single-sentence texts in our training corpus. In addition, because this Denoising Auto-Encoder (DAE) is much faster and more stable to train than a BE/S model, we train a standalone DAE that meaningfully outperforms our BE/S model and with highly readable summaries.

To further improve the performance of the DAE,

Figure 3: Denoising Auto-Encoding with Additive Sampling

we also train a separate version where we provide the decoder with the InferSent representation of the reference sentence in its initial hidden state. This provides the model with additional information to reconstruct the original sentence.

3.7 Training

In total, we train and evaluate three separate models. First, we have the BE/S model, which randomly alternates between BE/S and auto-encoding phases. Secondly, we train a standalone DAE model, which reconstructs the reference sentence given a noised set of reference words. Lastly, we train a modified DAE model that is provided and Infersent embedding of the reference sentence. We illustrate the complete training regime of BE/S in Figure 4. The DAE training, which follows a standard auto-encoding setup, is largely illustrated by Figure 3.

4 Data

We train and evaluate our models on the Annotated Gigaword (Napoles et al., 2012) dataset, which is a standard text summarization data set. However, only the reference texts are used in training, and the summaries are only ever used in evaluation. One disclaimer regarding using Gigaword is that the reference and summary domains are at least somewhat similar, with both being a single sentence, albeit of different lengths and with different levels of detail. This may not hold in other summarization data sets.

5 Results

In Table 1, We evaluate our models based on ROUGE (Lin, 2004) F1 scores, where a higher score is better. We provide comparisons with several baselines, including scoring the reference text itself and only using the first or last 10 words. We include a baseline that keeps the 10 least-common words while retaining order – this roughly extracts the keywords from the reference text. We also provide scores of supervised neural text-summarization methods from literature on Gigaword[1]. We also report the ROUGE scores scoring the generated summaries against the reference texts themselves – this is a proxy measure for how extractive the models are.

At a glance, we see that our three unsupervised models do not necessarily outperform the relevant baselines, especially First-10. We note that First-10 is a very strong baseline, getting comparable scores to even supervised models. The strong divergence between the First-10 and Last-10 baselines further demonstrates the extent to which the Gigaword dataset tends front-load the key information in its sentences. We also note that our generated summaries are longer than that of our baselines. This is because our models are trained on and therefore tend to output full sentences, whereas the true summaries and the baselines tend to have either sentence fragments or incomplete sentences. Note that the length of generated summaries is user-defined at inference time. We chose to target half the length of the reference text to maintain readability in generated summaries.

Among our models, DAE+NLI performs the best in ROUGE, reaching R1 close to that of First-10. DAE also produces good ROUGE scores, while BE/S, despite its more complex setup and its tendency to copy, is not up to par. Finally, the ROUGE scores with regards to the reference highlight that our models nearly completely rely on the reference vocabulary (as emphasized by the high R1) but also perform some rewording and sentence reformulations, as shown by the lower R2 scores against the reference compared to the other baselines (right side of table).

However, ROUGE is an imperfect metric for summarization, as word / n-gram overlap does not fully capture summary relevancy and meaningfulness[2]. Furthermore, the targets in Gigaword are

[1]F1 scores for (Rush et al., 2015) were not reported in the original paper, but where reported in (Nallapati et al., 2016) who obtained them from Rush

[2]See discussion in (Nallapati et al., 2016) or in (Paulus et al., 2017) where a reinforcement learning trained with a Rouge-L objective achieves the best scores but "produces the

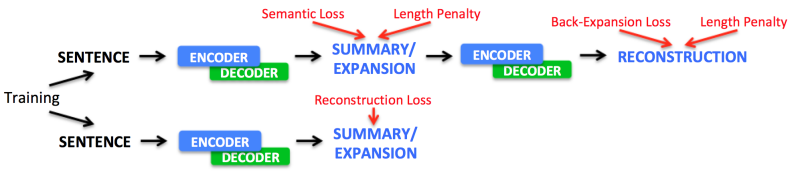

Figure 4: Back-Expansion / Summarization Training Regime

ROUGE on Annotated Gigaword							
	Target				Reference		
	R-1	R-2	R-L	*Avg. Length*	R-1	R-2	R-L
Baselines							
All text	28.91	10.22	25.08	31.3	100	100	100
First-10	**28.96**	**10.49**	**26.81**	10	47.31	44.32	47.31
Last-10	14.14	3.7	12.83	10	47.31	44.32	47.31
Uncommon-10	28.38	6.38	25.57	10	54.16	20.62	21.75
Unsupervised (Ours)							
BE/S	19.74	6.20	18.10	18.4	43.54	35.83	43.35
DAE	25.9	6.78	22.27	13.4	64.48	31.34	45.51
DAE + NLI	**28.27**	**7.55**	**24.34**	14.5	63.68	34.14	51.78
Supervised							
Rush et al. (2015)	29.78	11.89	26.97	-	-	-	-
Nallapati et al. (2016)	**32.67**	**15.59**	**30.64**	-	-	-	-

Table 1: ROUGE Scores for baselines, our unsupervised models, and supervised models from literature. BE/S=Back-Expansion / Summarization; DAE=Denoising Auto-Encoder; DAE+NLI=DAE with InferSent embeddings. Right half of table scores generated summaries against reference text, as opposed to ground-truth summaries.

headlines, which are worded more concisely than the sentences our models have seen. Therefore, we also conduct a qualitative evaluation to assess the quality of our models in producing legible and relevant summaries.

6 Error analysis

As can be seen in Figure 5 and 6, we see that summary quality does not directly correlate with ROUGE scores. Indeed, the model that produces the most grammatical and accurate sentences is the DAE, as shown in Examples 1 and 2 where the DAE produces semantically and linguistically accurate summaries of the reference text. Although the DAE+NLI model achieves higher ROUGE scores, it seems to produce less linguistic sentences and makes more semantic mistakes. This is surprising given the additional semantic information provided by the InferSent embedding, although there may be better ways of introducing that information. On the other hand, the BE/S model has trouble producing concise summaries and its emphasis on copying (imperfectly) means it produces few complete grammatical sentences.

Example 3 highlights the trouble our models have in dealing with longer sentences, where working with shuffled input makes inferring the correct word order and meaning more challenging. Example 4 shows how shuffling can lead our model to interchange two named entities for which it has insufficient information. It also shows a rare decoding artifact of our models where they will repeat a word several times, which beam search could possibly limit.

Overall, we see that DAEs are quite effective, but they deal poorly with the following conditions: (i) long sentences, (ii) named entities with little training data, (iii) text in references rarely used in summaries, such as long-form source citation ("said by X") (iv) more rarely, poor decoding conditioning. Fixing these is left for future work.

7 Discussion

We summarize our key takeaways from the above experiments:

Naive BE/S induces copying from the start/end. Our initial plan was for BE/S with gradients propagated through Gumbel-Softmax reparameterization to be the core of our unsupervised summarization approach, but we observed early on that models based on BE/S were particularly prone to simply copying from the front/back of the origin input sentence, resulting in ungrammatical sentences generated in the interim step. This was even more so the case as we started to rely on auto-

least readable summaries among [their] experiments"

Example 1:
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
F10: swedish truck maker volvo ab on tuesday reported its consecutive
L10: maker volvo ab on tuesday reported its consecutive quarterly weak
O10: detained iranian-american academic released from prison after hefty bail
BE/S: state-owned truck maker underwent ferrari on tuesday reported its third consecutive quarterly loss droughts year-on-year
DAE: volvo ab reported on tuesday its third consecutive quarterly loss .
DAE+NLI: swedish truck maker volvo ab on tuesday reported its consecutive quarterly weak .

Figure 5: Example for all baselines and models. **I** is the input, **G** (gold) is the true headline, **F10** and **L10** are the First 10 and Last 10 words, **O10** are the top 10 least common words in the original order.

encoding to better condition our text generation.

Without strong constraints on the interim outputs, an easy solution for the model to fall into is to simply copy the first N words from the input. Because the goal is to reconstruct the input, because the training regime involves applying the same model twice with different desired lengths, the models is able to reproduce at least the front of the sentence with a fairly high degree of accuracy. This adversely affects the ability of the model to learn any other more compressed or varied textual representation of the input. We also believe that copying from the back is also simple given the explicit length-countdown inputs, although copying from the front was far more prevalent.

NLI-based semantic loss is insufficient to induce correct semantics, but sentence embeddings are useful representations. We observed in training that the model could quickly learn to generate sentences that were scored 99% as "Entailments" from the input text, while either being wholly ungrammatical or unrelated to the input text. The fragility of neural-network based language models to perturbations in the input has also been shown in prior work (Jia and Liang, 2017). As such, we found the inclusion of the NLI-based loss to be unhelpful for later models. However, we observed that incorporating the Infersent embeddings in our DAE improved ROUGE scores, but produced somewhat less readable sentences. We believe that a refinement of this approach could lead to further improved results.

DAEs quickly learn to generate well-conditioned text even from badly conditioned inputs. We were surprised by the ability of DAEs to quickly learn to reorganize completely

Example 2:
I: a teenage girl student died and ## others , along with the driver of the coach , were injured early sunday in a single vehicle accident in tainan county , southern taiwan .
G: one dies ## injured in coach accident in tainan county
BE/S: convicted teenage girl student died and ## others , along with the driver of the coach year-on-year
DAE: a teenage girl died in the southern county of tainan early sunday .
DAE+NLI: a teenage girl and driver died of the vehicle accident in tainan county , taiwan .

Example 3:
I: under increasing pressure over the worsening cargo situation at the new airport , the government announced friday that an independent body would be set up to find out what went wrong and who 's to blame .
G: mounting losses for cargo handler ; government to start
BE/S: ceasefire increasing pressure over the worsening cargo situation at the new airport , the government announced friday that sentenced
DAE: the new government announced friday that the situation would be over and wrong .
DAE+NLI: increasing pressure over the cargo and an independent body at the airport 's worsening situation would find

Example 4:
I: li peng , chairman of the national people 's congress standing committee , met here today with ms. leung oi-sie , secretary of justice of the hong kong special administrative region -lrb- hksar -rrb- .
G: li peng meets hksar secretary of justice
BE/S: vice-premier vice-premier , chairman of the national people 's congress front-runner committee , met here today with xinhua vice-minister republic national national founding national people national national people
DAE: li peng , chairman of the hong kong special administrative region ms. ms. ms. .
DAE+NLI: li peng , chairman of the standing committee of the hong kong special administrative region congress .

Figure 6: More generated summaries

randomly shuffled input tokens into reasonably readable strings of English. Furthermore, even without conditioning on InferSent-based sentence-vectors, the DAEs were able to filter out the additively-sampled words with a fair degree of success. There are further ways we could augment the DAE model, such as learning specialized embeddings for named-entities in a reference text, or using more sophisticated models such as the Pointer-Generator.

8 Conclusion

We propose in this paper two strategies for unsupervised text summarization: Back-Expansion / Summarization and Denoising Auto-Encoders. We find DAEs to be far more effective than the BE/S, with the former being capable of generating fairly readable and semantically consistent summaries, while latter falls into an undesirable solution of largely copying the input. There remain many other strategies to be explored for improving unsupervised summarization, including the directly incorporation of language models, adversarial training, and textual style-transfer, that we hope to explore in future work.

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *CoRR*, abs/1710.11041.

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

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

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.

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

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.

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

Jiatao Gu, Daniel Jiwoong Im, and Victor O. K. Li. 2017. Neural machine translation with gumbel-greedy decoding. *CoRR*, abs/1706.07518.

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

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

E. Jang, S. Gu, and B. Poole. 2016. Categorical Reparameterization with Gumbel-Softmax. *ArXiv e-prints*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328.

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. *CoRR*, abs/1711.00043.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, pages 3075–3081.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

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*, AKBC-WEKEX '12, pages 95–100, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. corr abs/1509.00685 (2015).

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

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

A Code

The GitHub repository for our code can be found here[3]. In addition, we would like to give special credit to **Mikel Artetxe**, whose implementation of Unsupervised Neural Machine Translation we based our implementation heavily on.

B Collaboration Statement

Thibault Fevry focused on length-control, language model-based losses, and DAE-based approaches. **Jason Phang** focused on back-summarization, auto-encoding, NLI-based losses,

[3]For non-digital readers: https://github.com/zphang/nlu_project_2018q1

and sentence-embeddings. Both members contributed significantly in terms of literature review, idea generation, code implementation and experiments.