MeanSum : A Neural Model for Unsupervised Multi-Document Abstractive Summarization

Eric Chu [*] [†] [1] **Peter J. Liu** [*] [2]

Abstract

Abstractive summarization has been studied using neural sequence transduction methods with datasets of large, paired document-summary examples. However, such datasets are rare and the models trained from them do not generalize to other domains. Recently, some progress has been made in learning sequence-to-sequence mappings with only unpaired examples. In our work, we consider the setting where there are only documents (product or business reviews) with no summaries provided, and propose an end-to-end, neural model architecture to perform unsupervised abstractive summarization. Our proposed model consists of an auto-encoder where the mean of the representations of the input reviews decodes to a reasonable summary-review while not relying on any review-specific features. We consider variants of the proposed architecture and perform an ablation study to show the importance of specific components. We show through automated metrics and human evaluation that the generated summaries are highly abstractive, fluent, relevant, and representative of the average sentiment of the input reviews. Finally, we collect a reference evaluation dataset and show that our model outperforms a strong extractive baseline.

1. Introduction

Supervised, neural sequence-transduction models have seen wide success in many language-related tasks such as translation (Wu et al., 2016; Vaswani et al., 2017) and speech-recognition (Chiu et al., 2017). In these two cases, the model training is typically focused on the translation of sentences or recognition of short utterances, for which there is an abundance of parallel data. The application of such models to longer sequences (multi-sentence documents or long audio) works reasonably well in production systems because the sequences can be naturally decomposed into the shorter ones the models are trained on and thus sequence-transduction can be done piece-meal.

Similar neural models have also been applied to abstractive summarization, where large numbers of document-summary pairs are used to generate news headlines (Rush et al., 2015) or bullet-points (Nallapati et al., 2016; See et al., 2017). Work in this vein has been extended by Liu et al. (2018) to the multi-document[1] case to produce Wikipedia article text from references documents.

However, unlike translation or speech recognition, adapting such summarization models to different types of documents without re-training is much less reasonable; for example, in general documents do not decompose into parts that look like news articles, nor can we expect our idea of saliency or desired writing style to correspond with that of particular news publishers. Re-training or at least fine-tuning such models on many in-domain document-summary pairs should be expected to get desirable performance. Unfortunately, it is very expensive to create a large parallel summarization corpus and the most common case in our experience is that we have many documents to summarize, but have few or no examples of summaries.

We side-step these difficulties by completely avoiding the need for example summaries. Although there has been previous work on extractive summarization without supervision, we describe, to our knowledge, the first end-to-end, neural-abstractive, unsupervised summarization model. Unlike recent approaches to unsupervised translation (Artetxe et al., 2017; Lample et al., 2017), we do not only assume there is no parallel data, but also assume no dataset of output sequences.

In this paper, we study the problem of abstractively summarizing multiple reviews about a business or product without any examples and apply our method to publically available

*Equal contribution. †Work done primarily while interning at Google Brain. [1]MIT Media Lab [2]Google Brain. Correspondence to: Eric Chu <echu@mit.edu>, Peter J. Liu <peterjliu@google.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

[1]Note multi-document here means multiple documents about the same topic.

Yelp[2] and Amazon reviews (McAuley et al., 2015). We describe an architecture for summarizing multiple reviews in the form of a single review and perform multiple ablation experiments to justify the architecture chosen. We also define proxy metrics to evaluate our generated summaries and tune our models without example summaries, although we collect a crowd-sourced, reference evaluation set of summaries for additional evaluation. We further conduct human evaluation experiments to show that the generated summaries are often fluent, relevant, and representative of the summarized reviews. The code is available online[3].

2. Proposed Model

The MeanSum model consists of two main components: (1) an auto-encoder module that learns representations for each review and constrains the generated summaries to be in the language domain, and (2) a summarization module that learns to generate summaries that are semantically similar to each of the input documents. These contribute a reconstruction loss and similarity loss, respectively. Both components contain an LSTM encoder and decoder – the two encoders' weights are tied, and the two decoders' weights are tied. The encoder and decoder are also initialized with the same pre-trained language model trained on the reviews of the dataset. The overall architecture is shown in Figure 1.

Suppose we have an invertible tokenizer, T, that maps text documents, $\mathbb{D}$, to sequences of tokens (from a fixed vocabulary), $T(\mathbb{D})$. Let $\mathbb{V} \subset T(\mathbb{D})$ represent the tokenized reviews in our dataset with a maximum length of L. Given a set of k reviews about an entity (business or product), $\{x_1, x_2, ..., x_k\} \subset \mathbb{V}$, we would like to produce a document tokenized using the same vocabulary, $s \in T(\mathbb{D})$, that summarizes them.

In the auto-encoder sub-module, an encoder $\phi_E : \mathbb{V} \mapsto \mathbb{R}^n$, maps reviews to real-vector codes, $z_j = \phi_E(x_j)$. $\phi_E(x) = [h, c]$ is implemented as the concatenation of the final hidden and cell states of an LSTM (Hochreiter & Schmidhuber, 1997) after processing x one token at a time. A second decoder LSTM defines a distribution over $\mathbb{V}$ conditioned on the latent code, $p(x|z_j) = \phi_D(z_j)$, by initializing its state with z_j, and is trained using teacher-forcing (Williams & Zipser, 1989) with a standard cross-entropy loss to reconstruct the original reviews, i.e. the auto-encoder is implemented as a sequence-to-sequence model (Sutskever et al., 2014).

$$\ell_{rec}(\{x_1, x_2, ..., x_k\}, \phi_E, \phi_D) = \sum_{j=1}^{k} \ell_{cross\_entropy}(x_j, \phi_D(\phi_E(x_j))) \quad (1)$$

[2]https://www.yelp.com/dataset/challenge

[3]https://github.com/sosuperic/MeanSum

In the summarization module, $\{z_1, z_2, ..., z_k\}$ are combined using a simple mean over the hidden and cell states, $\bar{z} = [\bar{h}, \bar{c}]$, which is decoded by ϕ_D into the summary s. By using the same decoder as the auto-encoder, ϕ_D, we constrain the output summary to the space of reviews, $s \in \mathbb{V}$, and can think of it as a *canonical review*. We then re-encode the summary and compute a similarity loss that further constrains the summary to be semantically similar to the original reviews; we use average cosine distance, d_{cos}, between the hidden states h_j of each encoded review and h_s of the encoded summary, $\phi_E(s) = [h_s, c_s]$.

$$s \sim \phi_D(\bar{z}) \quad (2)$$

$$\ell_{sim}(\{x_1, x_2, ..., x_k\}, \phi_E, \phi_D) = \frac{1}{k} \sum_{j=1}^{k} d_{cos}(h_j, h_s) \quad (3)$$

As we lack ground truth summaries, we cannot use teacher forcing to generate the summary in Equation (2). Instead, we generate the summary using the Straight Through Gumbel-Softmax trick (Jang et al., 2016; Maddison et al., 2016), which approximates sampling from a categorical distribution (in this case a softmax over the vocabulary) and allows gradients to be backpropagated through this discrete generation process. We note that this sampling procedure allows us to avoid the exposure bias (Ranzato et al., 2015) of teacher-forcing, as the summary is generated through the same procedure during training and as in inference.

The final loss we optimize is simply $\ell_{model} = \ell_{rec} + \ell_{sim}$. We explored non-equal weighting of the losses but did not find a meaningful difference in outcomes.

2.1. Model Variations

To investigate the importance of different components and features, we evaluated the following variations of our model. Diagrams for the variants can be found in Appendix A:

No pre-trained language model. Instead of initializing the encoders and decoders with the weights of a pre-trained language model, the entire model was trained from scratch.

No auto-encoder. To test our belief that the auto-encoder is critical for (a) keeping the summaries in the review-language domain, $\mathbb{V}$, and (b) producing review representations that actually reflect the review, we tested a variant without the auto-encoder.

Reconstruction cycle loss. A perhaps more straightforward model architecture would be to encode the reviews, compute $\bar{z}$, generate the summary s, and then use s to decode into the reconstructed reviews $\hat{x}^j$, which would be used in a reconstruction loss with the original reviews. This last step would enforce the same constraint as the auto-encoding loss and the cosine similarity loss.

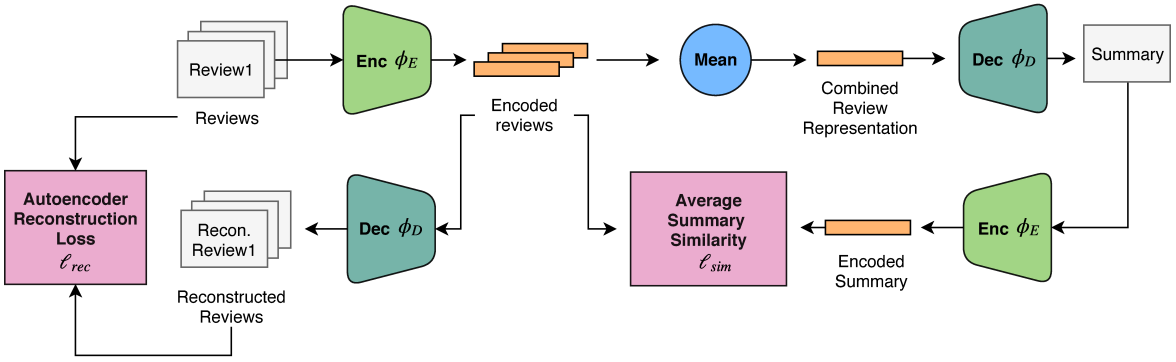

Figure 1. The proposed MeanSum model architecture.

Early cosine loss. In the similarity loss, instead of computing distance between encoded reviews and the encoded summary, $\phi_E(s)$, we use $\bar{z}$:

$$\ell_{sim}(\{x_1, x_2, ..., x_k\}, \phi_E, \phi_D) = \frac{1}{k}\sum_{j=1}^{k} d_{cos}(z_j, \bar{z}) \quad (4)$$

Perhaps this alone would be enough to push $\bar{z}$ into a latent space suitable for decoding into a summary. This would also preclude the need for back-propagating gradients through the discrete sampling step – we only need to decode the summary at test time, which we do through greedy decoding. In contrast to our model, summary generation here suffers from the exposure bias of teacher-forcing.

Untied decoders/encoders. We relax the constraint that the review and summary decoders/encoders share weights.

3. Metrics and Evaluation

3.1. Automated Metrics Without Summaries

Although we collected an evaluation set of reference summaries for the Yelp dataset described in section 3.2, it is useful to be able to tune models without having access to it, since creating summaries is labor-intensive. In our experiments we did not tune our models using the reference summaries at all and only used the following automatic statistics to guide model development.

Sentiment accuracy. A useful summary should reflect and be consistent with the overall sentiment of the reviews. We first separately train a CNN-based classification model that given a review x, predicts the star rating of a review, an integer from 1 to 5. For each summary, we check whether the classifier's max predicted rating is equal to the average rating of the reviews being summarized (rounded to the nearest star rating). This binary accuracy is averaged across all the data points:

$$\frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{CLF}(s^{(i)}) = round\Big(\frac{1}{k}\sum_{j=1}^{k} r(x_j^{(i)})\Big)\right] \quad (5)$$

where N is the number of data points, $\mathbf{CLF}$ is the trained classifier, $\mathbf{CLF}(s^{(i)})$ is the rating with the highest predicted probability, $s^{(i)}$ is the summary for the i-th data point, and $r(x_j^{(i)})$ is the rating for the j-th review in the i-th data point.

Word Overlap (WO) score. It is possible that a summary has the appropriate sentiment but is not grounded in information found in the reviews. As a sanity check that the summary is on-topic, we compute a measure of word overlap using the ROUGE-1 score (Lin, 2004) between the summary and each review and then average these scores, as shown in Equation 6. ROUGE is typically used between a candidate summary and a reference summary, but its use here similarly captures how much the candidate summary encapsulates the original documents. We note that in our use case, this metric is highly biased towards extractive summaries, as the "reference" is the original reviews themselves and maximizing it is not necessarily appropriate; however, very little word overlap is likely pathological.

$$\text{Word Overlap} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{k}\sum_{j=1}^{k}\mathbf{ROUGE}(s^{(i)}, x_j^{(i)})\right] \quad (6)$$

Negative Log-Likelihood (NLL). Generated summaries should also be fluent language. To measure this, we compute the negative log-likelihood of the summary according to a language model trained on the reviews. This metric is used to compare the outputs from different variations of our abstractive model.

3.2. Automated Evaluation With Reference Summaries

To validate our model selection process without reference summaries, we collected a set of 200 abstractive reference summaries for a subset of the Yelp dataset using Mechanical Turk[4]. We split this into 100 validation and 100 test examples for the benefit of future work, however we did not use either for model-tuning. As is customary in the summarization literature, we report ROUGE-F (Lin, 2004) scores between automatically generated and reference summaries. The collection process is described in Appendix B and an example can been seen in Figure E.1. Although we show ROUGE-2 results, they are arguably the least appropriate for abstractive summarization where rephrasing is common and encouraged.

3.3. Human Evaluation of Quality

To further assess the quality of summaries , we ran Mechanical Turk experiments (more details in Appendix B) asking workers to rate 100 summaries on a scale of 1 (very poor) to 5 (very good):

1. how well the sentiment of the summary agrees with the overall sentiment of the original review;
2. how well information is summarized across reviews;
3. the fluency of the summary based on five dimensions previously used in DUC-2005 (Dang, 2005): Grammaticality, Non-redundancy, Referential clarity, Focus, and Structure and Coherence.

4. Baseline Models

No training. We report the results of using the proposed architecture before optimizing ℓ_{model} to show that the quality of summaries improves beyond initializing with a pre-trained language model. As in the full model, the summary is generated by encoding the original reviews with ϕ_E, computing the combined review representation, and decoding the summary using ϕ_D.

Extractive. We use a recent, near state-of-the-art, centroid-based multi-document summarization method that uses word embeddings (Mikolov et al., 2013) instead of TF-IDF to represent each sentence (Rossiello et al., 2017). The maximum length of summaries was set to the 99.5^{th} percentile of reviews less than length L. This was chosen after evaluating various ceilings (e.g. 75^{th} percentile, 90^{th} percentile, no ceiling) on the validation set.

Best Review. It could be the case that one of the reviews would be a good summary. We thus compute the WO scores using each review as a summary. The review with the highest average (not including itself) WO score is selected.

[4]https://www.mturk.com

Worst Review. We use the same procedure as the Best Review baseline, except we select the review with the lowest average WO score to get an idea of what is a bad word overlap score.

Multi-Lead-1. The Lead-m baseline is often a strong baseline in single document summarization tasks and consists of the first m sentences in the document (See et al., 2017). We create an analog by first randomly shuffling the reviews, and then adding the first sentence from each review until the maximum length L is reached. If L is not reached, then the summary is composed of the first sentence from each review.

5. Related Work

Many popular extractive summarization techniques do not require example summaries and instead consider summarization as a sentence-selection problem. Sentences may be selected based on scores computed from the presence of topic-words or word-frequencies (Nenkova & Vanderwende, 2005). Centroid-based methods try to select sentences such that the resulting summary is close to the centroid of the input documents in the representation space (Radev et al., 2004). Rossiello et al. (2017) extend this approach by mapping sentences to their representation using word2vec embeddings rather than using TF-IDF weights. The main disadvantage of extractive methods is their limitation in copying text from the input, which is not how humans summarize. Banko & Vanderwende (2004) in particular found human-authored summaries of multiple documents to be much more abstractive.

Liu et al. (2015) present a framework for doing abstractive summarization in three stages. First text is parsed to an Abstract Meaning Representation (AMR) graph; then a graph-summarization procedure is carried out, which extracts an AMR sub-graph; finally, text is generated from this sub-graph. All three components require separate training and also AMR annotations, for which there is very little data. It is unclear how to generalize this to the multi-document setting.

Miao & Blunsom (2016) train an auto-encoder to do extractive sentence compression and combines it with a model trained on parallel data to do semi-supervised summarization.

Recently, there has been progress in learning to translate between languages using only unpaired example sentences from each language (Artetxe et al., 2017; Lample et al., 2017). Gomez et al. (2018) and Wang & Lee (2018) train CycleGan-like (Zhu et al., 2017) models to map between unpaired examples of (cipher-text, decrypted-text) and (article, headline), respectively. In contrast to this line of interesting work, we only have examples of the input sequence and thus

cannot apply such techniques.

Review summarization systems have been designed with domain-specific choices. Liu et al. (2005); Ly et al. (2011) focus on producing a highly-structured summary consisting of facets and example positive and negative sentences for each. Zhuang et al. (2006) incorporate external databases to construct simiarly structured, movie-specific review summaries. In contrast our summaries have no explicit constraints or templates.

6. Experimental Setup

6.1. Datasets

We tuned our models primarily on a dataset of customer reviews provided in the Yelp Dataset Challenge, where each review is accompanied by a 5-star rating. We used a data-driven, wordpiece tokenizer (Wu et al., 2016) with a vocabulary size of 32,000 and filtered reviews to those with tokenized length, $L \leq 150$. Businesses were then filtered to those with at least 50 reviews, so that every business had enough reviews to be summarized. Finally, we removed businesses above the 90^{th} percentile in review count in order to prevent the dataset from being dominated by a small percent of hugely popular businesses. The final training, validation, and test splits consist of 10695, 1337, and 1337 businesses, and 1038184, 129856, and 129840 reviews, respectively.

6.2. Experimental Details

The language model, encoders, and decoders were multiplicative LSTM's (Krause et al., 2016) with 512 hidden units, a 0.1 dropout rate, a word embedding size of 256, and layer normalization (Ba et al., 2016). We used Adam (Kingma & Ba, 2014) to train, a learning rate of 0.001 for the language model, a learning rate of 0.0001 for the classifier, and a learning rate of 0.0005 for the summarization model, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial temperature for the Gumbel-softmax was set to 2.0.

One input item to the language model was $k = 8$ reviews from the same business or product concatenated together with end-of-review delimiters, with each update step operating on a subsequence of 256 subtokens. The initial states were set to zero and persisted across update steps for that set of $k = 8$ reviews in order to simulate full back-propagation through the entire sequence. The review-rating classifier was a multi-channel text convolutional neural network similar to Kim (2014) with 3,4,5 width filters, 128 feature maps per filter, and a 0.5 dropout rate. The classifier achieves 72% accuracy, which is similar to current state-of-the-art performance on the Yelp dataset.

7. Results

7.1. Main Results

The automated metrics for our model and the baselines are shown in Table 1. Using the final test split of the collected reference summaries, we find that our model obtains at least comparable ROUGE scores to all the baselines, and outperforms the extractive model signifcantly with with p-values (Wilcoxon test) of 2.102×10^{-7}, 0.0217, and 2.471×10^{-6}. These findings are also consistent across 100 random validation-test splits on the 200 reference summaries – the scores on the final validation set and the mean and standard deviation of scores across these trials are found in Table C.1 and Table C.2.

The abstractive model outperforms all the baselines in sentiment accuracy. It also obtains a slightly lower, but comparable Word Overlap compared to the extractive method. Although it is a proxy for ROUGE with references, the Word Overlap scores are correlated with ROUGE-1 and ROUGE-L, with statistically significant pearson coefficients of 0.797 and 0.728, respectively. This suggests word overlap is a reasonable metric for guiding model development in the absence of an evaluation set of reference summaries.

The human evaluation results on the quality of the generated summaries are shown in Table 2. The human scores on sentiment and information agreement are comparable for the extractive and abstractive models and rank order the methods similarly to our proxy metrics: sentiment accuracy and word overlap, respectively. We find that the abstractive summaries are comparable to extractive summaries and randomly selected input reviews on fluency, suggesting the model outputs have high fluency. We also find that the early cosine loss model has much lower ratings on the fluency questions, which agrees with the much higher NLL compared to our best-performing abstractive model.

An example set of reviews and corresponding summaries are shown in Figure 2. We find that extractive summaries, while highly specific and fluent, appear to summarize only a subset of the reviews. The abstractive summaries tend to be more general (e.g. using the term "mani/pedi" which does not occur in the input), but as the higher sentiment accuracy suggests, also more reflective of the average sentiment of the reviews. Because the model has no attention, there is very little copying and the summaries are highly abstractive. For summaries over the test set, 78.43% of 2-grams, 96.57% of 3-grams, and 99.33% of 4-grams in the summaries are unique (i.e. not found in the reviews being summarized). Figure 3 shows summaries of negative, neutral, and positive reviews from the same business, allowing us to see how the summary changes with the input sentiment. An example with the reference summary is shown in Figure E.1, and more examples can be found in Appendix E.

Table 1. Automated metric results with $k = 8$ reviews being summarized. The reference summaries results are shown for the test split. Note for Best/Worst Review WO scores: we exclude the best/worst review when calculating the average. Numbers are not provided for models that degenerated into non-natural language. As noted earlier, the NLL's are only provided for our abstractive models.

		Vs. Reference Summaries			*Metrics Without Summaries*		
	Model	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**Sentiment Acc.**	**WO**	**NLL**
	MeanSum (ours)	28.86	3.66	15.91	**51.75**	26.09	1.19
Baselines	Extractive (Rossiello et al., 2017)	24.61	2.85	13.81	42.95	28.59	–
	No training	21.22	1.69	11.92	24.44	19.68	1.29
	Best review	27.97	3.46	15.29	38.48	23.86	–
	Worst review	16.91	1.66	11.11	30.01	13.14	–
	Multi-Lead-1	26.79	**3.77**	14.39	40.69	**31.64**	–
Model Variants	No pre-trained language model	26.16	3.07	15.31	48.97	23.67	**1.14**
	No auto-encoder	–	–	–	–	–	–
	Reconstruction cycle loss	25.23	3.58	15.82	43.65	22.26	**1.14**
	Early cosine loss	14.35	1.26	9.02	19.32	14.28	1.71
	Untied decoders	–	–	–	–	–	–
	Untied encoders	**29.35**	3.52	**15.97**	50.89	26.29	1.20

Table 2. Mechanical Turk results evaluating quality of summaries.

Model	**Sentiment**	**Information**
MeanSum (ours)	**3.91**	3.83
Extractive	3.87	**3.85**

Model	**Grammar**	**Non-redundancy**	**Referential clarity**	**Focus**	**Structure and Coherence**
MeanSum (ours)	**3.97**	3.74	**4.13**	4.10	**4.02**
Extractive	3.86	3.93	4.05	4.01	3.99
Early cosine loss	2.02	1.84	2.02	1.96	1.95
Random review	3.94	**4.06**	4.09	**4.23**	4.01

7.2. Model Variant Ablation Studies

The results of the ablation studies are shown in Table 1.

The language model experiments indicate that while initializing the weights with a pre-trained language model helps, as has been shown in sequence-to-sequence models (Ramachandran et al., 2016), it is not critical. Without the pre-trained language model, the metrics are only a few points lower – ROUGE-1 (26.16 vs. 28.86), sentiment accuracy (51.75 vs. 48.97), WO score (26.09 vs. 23.86). We also find that using a pre-trained language model alone, without training the summarization model, is not enough to generate good summaries. While the generated texts are fluent, they fail to actually summarize the reviews, as shown by the low ROUGE (ROUGE-1 = 21.22), sentiment accuracy (24.44), and WO score (19.68).

Two of the models failed completely, with the model degenerating from producing natural language (even though initialized with the pre-trained language model) to garbage text. The first variant, without the auto-encoder, converges to a trivial solution – the cosine similarity loss can be minimized if the encoders learn to produce the same representation z_j regardless of the input. As suspected with the second variant, in which the decoders are not tied, the summary decoder has no constraint to remain in language space, $\mathbb{V}$.

The reconstruction cycle loss works, but worse than the original model – ROUGE-1 (25.23 vs. 28.86), sentiment accuracy (43.66 vs. 51.75), WO score (22.26 vs. 26.09). We hypothesize its lower performance is due to difficulties in optimization. Although the Gumbel softmax trick allows the model to be fully differentiable, the gradients will have

Original Reviews: Mean Rating = 4
No question the best pedicure in Las Vegas. I go around the world to places like Thailand and Vietnam to get beauty services and this place is the real thing. Ben, Nancy and Jackie took the time to do it right and you don't feel rushed. My cracked heels have never been softer thanks to Nancy and they didn't hurt the next day. </DOC> Came to Vegas to visit sister both wanted full sets got to the salon like around 4 . Friendly guy greet us and ask what we wanted for today but girl doing nails was very rude and immediately refuse service saying she didn't have any time to do 2 full sets when it clearly said open until 7pm! </DOC> This is the most clean nail studio I have been so far. The service is great. They take their time and do the irk with love. That creates a very comfortable atmosphere. I recommend it to everyone!! </DOC> Took a taxi here from hotel bc of reviews -Walked in and walked out - not sure how they got these reviews. Strong smell and broken floor - below standards for a beauty care facility. </DOC> The best place for pedi in Vegas for sure. My husband and me moved here a few months ago and we have tried a few places, but this is the only place that makes us 100% happy with the result. I highly recommend it! </DOC> This was the best nail experience that I had in awhile. The service was perfect from start to finish! I came to Vegas and needed my nails, feet, eyebrows and lashes done before going out. In order to get me out quickly, my feet and hands where done at the same time. Everything about this place was excellent! I will certainly keep them in mind on my next trip. </DOC> I came here for a munch needed pedicure for me and my husband. We got great customer service and an amazing pedicure and manicure. I will be back every time I come to Vegas. My nails are beautiful, my skin is very soft and smooth, and most important I felt great after leaving!!! </DOC> My friend brought me here to get my very first manicure for my birthday. Ben and Nancy were so friendly and super attentive. Even though were were there past closing time, I never felt like we were being rushed or that they were trying to get us out the door. I got the #428 Rosewood gel manicure and I love it. I'll definitely be back and next time I'll try a pedicure.

Extractive Summary: Predicted Rating = 1
Came to Vegas to visit sister both wanted full sets got to the salon like around 4 . Friendly guy greet us and ask what we wanted for today but girl doing nails was very rude and immediately refuse service saying she didn't have any time to do 2 full sets when it clearly said open until 7pm!

Unsupervised Abstractive Summary: Predicted Rating = 5
Probably the best mani/pedi I have ever had. I went on a Saturday afternoon and it was busy and they have a great selection of colors. We went to the salon for a few hours of work, but this place was very relaxing. Very friendly staff and a great place to relax after a long day of work.

Figure 2. An example of input Yelp Reviews (separated by "</DOC>") with the extractive baseline and our model summaries. Certain words are colored in the original reviews that correspond to similar words in the abstractive summary.

Summary of Negative Reviews: Predicted Rating = 1
Never going back. Went there for a late lunch and the place was packed with people. I had to ask for a refund, a manager was rude to me and said they didn't have any. It's not the cheapest place in town but it's not worth it for me. And they do not accept debit cards no matter how busy it is. But whatever, they deserve the money .

Summary of Neutral Reviews: Predicted Rating = 3
Food is good and the staff was friendly. I had the pulled pork tacos, which was a nice surprise. The food is not bad but certainly not great. Service was good and friendly. I would have given it a 3 star but I'm not a fan of their food. Service was friendly and attentive. Only complaint is that the staff has no idea what he's talking about, but it's a little more expensive than other taco shops.

Summary of Positive Reviews: Predicted Rating = 5
Always great food. The best part is that it's on the light rail station, and it's a little more expensive than most places. I had a brisket taco with a side of fries and a side of corn. Great place to take a date or to go with some friends

Figure 3. Summaries generated from our model for one business, but varying the input reviews. Each summary is for a set of reviews with the same rating. The original reviews are found in Appendix E.3

either high bias or or variance depending on the temperature (which can be annealed during training). With the single loss function being after the Gumbel softmax step, the model may be difficult to optimize. We also believe that reconstructing the original texts from $\phi_E(s)$ is difficult, as s is a lossy compression of the original documents.

Next, we see that the "Early cosine loss" model has poor sentiment accuracy and average Word Overlap. We believe this is largely due to exposure bias, resulting in a relatively large NLL. The summaries are generated at test time through greedy decoding, but this decoding process (and thus the summary decoder) is not part of the training procedure. Critically, the full model does not suffer from this problem. Manual inspection of the samples confirm the summaries are disfluent.

Finally, the model performed approximately as well without tying the encoders – ROUGE-1 (29.35 vs. 28.86), sentiment accuracy (50.89 vs. 51.75), WO score (26.29 vs. 26.09 WO). Given that this is the case, it makes sense to simply tie the encoders and reduce the number of model parameters.

We also examined the fluency of each model by plotting the negative log-likelihood of the generated summaries during training, as shown in Figure 4. The two models that fail are immediately evident, as indicated by their large NLL's.

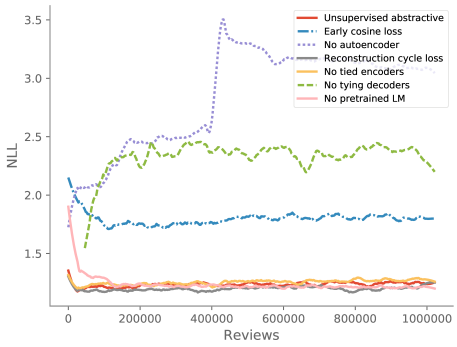

Figure 4. Negative log-likelihood of models during training

7.3. Varying Number of Input Documents, k

Because our model can encode input documents (reviews) in parallel and the mean operation over encodings is trivial, it is highly scalable in the number of input reviews k. To examine the robustness with respect to k on quality, we ran experiments varying this hyperparameter. The ROUGE scores on the reference summaries are shown in Table 3, indicating that the scores are largely consistent across different k's at train time. The *varying-k* model was trained with $k \in \{4, 8, 16\}$ reviews (randomly selected every minibatch). We did not perform any extra hyperparameter search for the models trained with $k = 4$, $k = 16$, and *varying-k*, and instead used the same hyperparmeters used to train the $k = 8$ model. The results for the other automated metrics are shown in Appendix Figures D.1 and D.2, and also support the finding that our model is relatively robust to k.

Table 3. ROUGE vs reference summaries and different k's at train time

Model	ROUGE-1	ROUGE-2	ROUGE-L
$k = 4$	27.30	3.23	16.10
$k = 8$	28.86	3.66	3.66
$k = 16$	28.25	3.21	16.00
varying-k	27.09	2.85	14.95

7.4. Amazon Dataset

To check that our model works beyond Yelp reviews, we also tested on a Amazon dataset of product reviews. We selected two different categories – Movies & TV and Electronics. We used the same parameters used to filter the Yelp dataset, resulting in training, validation, and test splits of 6,237, 780, and 780 products, and 583776, 73040, 73040 reviews, respectively. No tuning was performed – we used the exact same model and training hyperparameters as the Yelp models. The automatic metrics without references are shown in Table 4. We find similar results – the abstractive method outperforms the baselines in sentiment accuracy and has slightly lower Word Overlap than the extractive baseline method.

Table 4. Results on Amazon dataset

Model	Sentiment Acc.	WO	NLL
MeanSum (ours)	**47.90**	27.02	**1.23**
Extractive	43.86	30.41	1.38
No Training	38.04	18.10	1.37
Best review	45.05	24.59	1.29
Worst review	38.88	13.79	1.36
Multi-Lead-1	44.90	**32.18**	1.33

7.5. Qualitative Error Analysis

Although most summaries look reasonable, there are occasionally failure modes. The commonerrors are a) fluency problems, b) factual inaccuracy (e.g. the incorrect city is referenced), c) poorer performance on categories with limited data, and d) contradictory reviews (positive statement followed by a negative statement about the same subject). More discussion and examples are in Appendix E.

8. Conclusion, Limitations, and Future Work

The standard approaches to neural abstractive summarization use supervised learning with many document-summary pairs that are expensive to obtain at scale. To address this limitation and make progress toward more widely useful models, we introduced an unsupervised abstractive model for multi-document summarization that applied to reviews is competitive with unsupervised extractive methods.

The proposed model is highly abstractive because it lacks attention or pointers – future work could incorporate these mechanisms to provide summaries that contain both the most relevant points and the specific details to support them.

For our problem, summarizing multiple documents in the form of a similarly distributed single-document was appropriate, but may not be in all multi-document summarization cases. Learning to tailor the summary to a different desired form (with few examples) would be an interesting extension.

Our model does not provide an unsupervised solution for the more difficult (as there are fewer redundancy cues) single-document summarization problem. Here too there are extractive solutions, but extending ideas presented in this paper might yield the similar advantages of neural abstractive summarization.

ACKNOWLEDGMENTS

We thank Kai Chen, Trieu Trinh, David Grangier, and Jie Ren for helpful comments on the manuscript, and Jeff Dean, Samy Bengio, Claire Cui, and Deb Roy for their support of this work.

References

Artetxe, M., Labaka, G., Agirre, E., and Cho, K. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*, 2017.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Banko, M. and Vanderwende, L. Using n-grams to understand the nature of summaries. In *Proceedings of HLT-NAACL 2004: Short Papers*, pp. 1–4. Association for Computational Linguistics, 2004.

Chiu, C.-C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, K., et al. State-of-the-art speech recognition with sequence-to-sequence models. *arXiv preprint arXiv:1712.01769*, 2017.

Dang, H. T. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pp. 1–12, 2005.

Gomez, A. N., Huang, S., Zhang, I., Li, B. M., Osama, M., and Kaiser, L. Unsupervised cipher cracking using discrete gans. *arXiv preprint arXiv:1801.04883*, 2018.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krause, B., Lu, L., Murray, I., and Renals, S. Multiplicative lstm for sequence modelling. *arXiv preprint arXiv:1609.07959*, 2016.

Lample, G., Denoyer, L., and Ranzato, M. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

Liu, B., Hu, M., and Cheng, J. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pp. 342–351. ACM, 2005.

Liu, F., Flanigan, J., Thomson, S., Sadeh, N., and Smith, N. A. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1077–1086, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1114.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1704.04368*, 2018.

Ly, D. K., Sugiyama, K., Lin, Z., and Kan, M.-Y. Product review summarization from a deeper perspective. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pp. 311–314. ACM, 2011.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

McAuley, J., Targett, C., Shi, Q., and Van Den Hengel, A. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52. ACM, 2015.

Miao, Y. and Blunsom, P. Language as a latent variable: Discrete generative models for sentence compression. *arXiv preprint arXiv:1609.07317*, 2016.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Nallapati, R., Zhou, B., dos Santos, C., glar Gulçehre, Ç., and Xiang, B. Abstractive text summarization using sequence-to-sequence rnns and beyond. *CoNLL 2016*, pp. 280, 2016.

Nenkova, A. and Vanderwende, L. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101, 2005.

Radev, D. R., Jing, H., Styś, M., and Tam, D. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

Ramachandran, P., Liu, P. J., and Le, Q. V. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*, 2016.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Rossiello, G., Basile, P., and Semeraro, G. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pp. 12–21, 2017.

Rush, A. M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 379–389, 2015. URL http://aclweb.org/anthology/D/D15/D15-1044.pdf.

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

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Wang, Y. and Lee, H.-y. Learning to encode text as human-readable summaries using generative adversarial networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4187–4195. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/D18-1451.

Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2242–2251. IEEE, 2017.

Zhuang, L., Jing, F., and Zhu, X.-Y. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pp. 43–50. ACM, 2006.