Simple Unsupervised Summarization by Contextual Matching

Jiawei Zhou
Harvard University
jzhou02@g.harvard.edu

Alexander M. Rush
Harvard University
srush@seas.harvard.edu

Abstract

We propose an unsupervised method for sentence summarization using only language modeling. The approach employs two language models, one that is generic (i.e. pretrained), and the other that is specific to the target domain. We show that by using a product-of-experts criteria these are enough for maintaining continuous contextual matching while maintaining output fluency. Experiments on both abstractive and extractive sentence summarization data sets show promising results of our method without being exposed to any paired data.

1 Introduction

Automatic text summarization is the process of formulating a shorter output text than the original while capturing its core meaning. We study the problem of unsupervised sentence summarization with no paired examples. While data-driven approaches have achieved great success based on various powerful learning frameworks such as sequence-to-sequence models with attention (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016), variational auto-encoders (Miao and Blunsom, 2016), and reinforcement learning (Paulus et al., 2017), they usually require a large amount of parallel data for supervision to do well. In comparison, the unsupervised approach reduces the human effort for collecting and annotating large amount of paired training data.

Recently researchers have begun to study the unsupervised sentence summarization tasks. These methods all use parameterized unsupervised learning methods to induce a latent variable model: for example Schumann (2018) uses a length controlled variational autoencoder, Fevry and Phang (2018) use a denoising autoencoder but only for extractive summarization, and Wang and Lee (2018) apply a reinforcement learning procedure combined with GANs, which takes a further step to the goal of Miao and Blunsom (2016) using language as latent representations for semi-supervised learning.

This work instead proposes a simple approach to this task that does not require any joint training. We utilize a generic pretrained language model to enforce contextual matching between sentence prefixes. We then use a smoothed problem specific target language model to guide the fluency of the generation process. We combine these two models in a product-of-experts objective. This approach does not require any task-specific training, yet experiments show results on par with or better than the best unsupervised systems while producing qualitatively fluent outputs. The key aspect of this technique is the use of a pretrained language model for unsupervised contextual matching, i.e. unsupervised paraphrasing.

2 Model Description

Intuitively, a sentence summary is a shorter sentence that covers the main point succinctly. It should satisfy the following two properties (similar to Pitler (2010)): (a) Faithfulness: the sequence is close to the original sentence in terms of meaning; (b) Fluency: the sequence is grammatical and sensible to the domain.

We propose to enforce the criteria using a product-of-experts model (Hinton, 2002),

$$\mathbf{P}(\mathbf{y}|\mathbf{x}) \propto p_{\text{cm}}(\mathbf{y}|\mathbf{x}) p_{\text{fm}}(\mathbf{y}|\mathbf{x})^{\lambda}, \quad |\mathbf{y}| \leq |\mathbf{x}| \quad (1)$$

where the left-hand side is the probability that a target sequence $\mathbf{y}$ is the summary of a source sequence $\mathbf{x}$, $p_{\text{cm}}(\mathbf{y}|\mathbf{x})$ measures the faithfulness in terms of contextual similarity from $\mathbf{y}$ to $\mathbf{x}$, and $p_{\text{fm}}(\mathbf{y}|\mathbf{x})$ measures the fluency of the token sequence $\mathbf{y}$ with respect to the target domain. We

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5101–5106
Florence, Italy, July 28 - August 2, 2019.

use λ as a hyper-parameter to balance the two expert models.

We consider this distribution (1) being defined over all possible $\mathbf{y}$ whose tokens are restricted to a candidate list C determined by $\mathbf{x}$. For extractive summarization, C is the set of word types in $\mathbf{x}$. For abstractive summarization, C consists of relevant word types to $\mathbf{x}$ by taking K closest word types from a full vocabulary V for each source token measured by pretrained embeddings.

2.1 Contextual Matching Model

The first expert, $p_{\text{cm}}(\mathbf{y}|\mathbf{x})$, tracks how close $\mathbf{y}$ is to the original input $\mathbf{x}$ in terms of a contextual "trajectory". We use a pretrained language model to define the left-contextual representations for both the source and target sequences. Define $S(x_{1:m}, y_{1:n})$ to be the contextual similarity between a source and target sequence of length m and n respectively under this model. We implement this as the cosine-similarity of a neural language model's final states with inputs $x_{1:m}$ and $y_{1:n}$. This approach relies heavily on the observed property that similar contextual sequences often correspond to paraphrases. If we can ensure close contextual matching, it will keep the output faithful to the original.

We use this similarity function to specify a generative process over the token sequence $\mathbf{y}$,

$$p_{\text{cm}}(\mathbf{y}|\mathbf{x}) = \prod_{n=1}^{N} q_{\text{cm}}(y_n|\mathbf{y}_{<n}, \mathbf{x}).$$

The generative process aligns each target word to a source prefix. At the first step, $n = 1$, we compute a greedy alignment score for each possible word $w \in \mathcal{C}$, $s_w = \max_{j \geq 1} S(x_{1:j}, w)$ for all source prefixes up to length j. The probability $q_{\text{cm}}(y_1 = w|\mathbf{x})$ is computed as $\text{softmax}(\mathbf{s})$ over all target words. We also store the aligned context $z_1 = \arg\max_{j \geq 1} S(x_{1:j}, y_1)$.

For future words, we ensure that the alignment is strictly monotonic increasing, such that $z_n < z_{n+1}$ for all n. Monotonicity is a common assumption in summarization (Yu et al., 2016a,b; Raffel et al., 2017). For $n > 1$ we compute the alignment score $s_w = \max_{j > z_{n-1}} S(x_{1:j}, [y_{1:n-1}, w])$ to only look at prefixes longer than z_{n-1}, the last greedy alignment. Since the distribution conditions on $\mathbf{y}$ the past alignments are deterministic to compute (and can be stored). The main computational cost is in extending the target language model context to compute S.

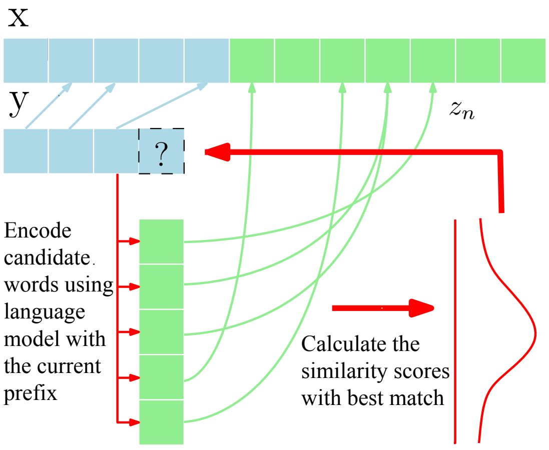
x
y
z_n
?
Encode candidate words using language model with the current prefix
Calculate the similarity scores with best match

Figure 1: Generative process of the contextual matching model.

This process is terminated when a sampled token in $\mathbf{y}$ is aligned to the end of the source sequence $\mathbf{x}$, and the strict monotonic increasing alignment constraint guarantees that the target sequence will not be longer than the source sequence. The generative process of the above model is illustrated in Fig. 1.

2.2 Domain Fluency Model

The second expert, $p_{\text{fm}}(\mathbf{y}|\mathbf{x})$, accounts for the fluency of $\mathbf{y}$ with respect to the target domain. It directly is based on a domain specific language model. Its role is to adapt the output to read closer shorter sentences common to the summarization domain. Note that unlike the contextual matching model where $\mathbf{y}$ explicitly depends on $\mathbf{x}$ in its generative process, in the domain fluency language model, the dependency of $\mathbf{y}$ on $\mathbf{x}$ is implicit through the candidate set C that is determined by the specific source sequence $\mathbf{x}$.

The main technical challenge is that the probabilities of a pretrained language model are not well-calibrated with the contextual matching model within the candidate set C, and so the language model tends to dominate the objective because it has much higher variance (more peaky) in the output distribution than the contextual matching model. To manage this issue we apply kernel smoothing over the language model to adapt it from the full vocab V down to the candidate word list C.

Our smoothing process focuses on the output embeddings from the pretrained language model. First we form the Voronoi partition (Aurenhammer, 1991) over all the embeddings using the candidate set C. That is, each word type w' in the full vocabulary V is exactly assigned to one region represented by a word type w in the candidate set C, such that the distance from w' to w is not greater than its distance to any other word types in C. As above, we use cosine similarity between corresponding word embeddings to define the regions. This results in a partition of the full vocabulary space into $|C|$ distinct regions, called Voronoi cells. For each word type $w \in C$, we define $\mathcal{N}(w)$ to be the Voronoi cell formed around it. We then use cluster smoothing to define a new probability distribution:

$$p_{\text{fm}}(\mathbf{y}|\mathbf{x}) = \prod_{n=1}^{N} \sum_{w' \in \mathcal{N}(y_n)} \text{lm}(w'|\mathbf{y}_{<n})$$

where lm is the conditional probability distribution of the pretrained domain fluency language model. By our construction, p_{fm} is a valid distribution over the candidate list C. The main benefit is that it redistributes probability mass lost to terms in V to the active words in C. We find this approach smoothing balances integration with p_{cm}.

2.3 Summary Generation

To generate summaries we maximize the log probability (1) to approximate $\mathbf{y}^*$ using beam search. We begin with a special start token. A sequence is moved out of beam if it has aligned to the end token appended to the source sequence. To discourage extremely short sequences, we apply length normalization to re-rank the finished hypotheses. We choose a simple length penalty as $lp(\mathbf{y}) = |\mathbf{y}| + \alpha$ with α a tuning parameter.

3 Experimental Setup

For the contextual matching model's similarity function S, we adopt the forward language model of ELMo (Peters et al., 2018) to encode tokens to corresponding hidden states in the sequence, resulting in a three-layer representation each of dimension 512. The bottom layer is a fixed character embedding layer, and the above two layers are LSTMs associated with the generic unsupervised language model trained on a large amount of text data. We explicitly manage the ELMo hidden states to allow our model to generate contextual embeddings sequentially for efficient beam search.[1] The fluency language model component lm is task specific, and pretrained on a corpus of summarizations. We use an LSTM model with 2 layers, both embedding size and hidden size set to 1024. It is trained using dropout rate 0.5 and SGD combined with gradient clipping.

We test our method on both abstractive and extractive sentence summarization tasks. For abstractive summarization, we use the English Gigaword data set pre-processed by Rush et al. (2015). We train p_{fm} using its 3.8 million headlines in the training set, and generate summaries for the input in test set. For extractive summarization, we use the Google data set from Filippova and Altun (2013). We train p_{fm} on 200K compressed sentences in the training set and test on the first 1000 pairs of evaluation set consistent with previous works. For generation, we set $\lambda = 0.11$ in (1) and beam size to 10. Each source sentence is tokenized and lowercased, with periods deleted and a special end of sentence token appended. In abstractive summarization, we use $K = 6$ in the candidate list and use the fixed embeddings at the bottom layer of ELMo language model for similarity. Larger K has only small impact on performance but makes the generation more expensive. The hyper-parameter α for length penalty ranges from -0.1 to 0.1 for different tasks, mainly for desired output length as we find ROUGE scores are not sensitive to it. We use concatenation of all ELMo layers as default in p_{cm}.

4 Results and Analysis

Quantitative Results. The automatic evaluation scores are presented in Table 1 and Table 2. For abstractive sentence summarization, we report the ROUGE F1 scores compared with baselines and previous unsupervised methods. Our method outperforms commonly used prefix baselines for this task which take the first 75 characters or 8 words of the source as a summary. Our system achieves comparable results to Wang and Lee (2018) a system based on both GANs and reinforcement training. Note that the GAN-based system needs both source and target sentences for training (they are unpaired), whereas our method only needs the target domain sentences for a simple language model. In Table 1, we also list scores of the state-of-the-art supervised model, an attention based

[1] Code available at https://github.com/jzhou316/Unsupervised-Sentence-Summarization.

Model	R1	R2	RL
Lead-75C	23.69	7.93	21.5
Lead-8	21.30	7.34	19.94
Schumann (2018)	22.19	4.56	19.88
Wang and Lee (2018)	27.09	9.86	24.97
Contextual Match	26.48	10.05	24.41
Cao et al. (2018)	37.04	19.03	34.46
seq2seq	33.50	15.85	31.44
Contextual Oracle	37.03	15.46	33.23

Table 1: Experimental results of abstractive summarization on Gigaword test set with ROUGE metric. The top section is prefix baselines, the second section is recent unsupervised methods and ours, the third section is state-of-the-art supervised method along with our implementation of a seq-to-seq model with attention, and the bottom section is our model's oracle performance. Wang and Lee (2018) is by author correspondence (scores differ because of evaluation setup). For another unsupervised work Fevry and Phang (2018), we attempted to replicate on our test set, but were unable to obtain results better than the baselines.

Model	F1	CR
F&A Unsupervised	52.3	-
Contextual Match	60.90	0.38
Filippova et al. (2015)	82.0	0.38
Zhao et al. (2018)	85.1	0.39

Table 2: Experimental results of extractive summarization on Google data set. F1 is the token overlapping score, and CR is the compression rate. F&A is an unsupervised baseline used in Filippova and Altun (2013), and the bottom section is supervised results.

seq-to-seq model of our own implementation, as well as the oracle scores of our method obtained by choosing the best summary among all finished hypothesis from beam search. The oracle scores are much higher, indicating that our unsupervised method does allow summaries of better quality, but with no supervision it is hard to pick them out with any unsupervised metric. For extractive sentence summarization, our method achieves good compression rate and significantly raises a previous unsupervised baseline on token level F1 score.

Analysis. Table 3 considers analysis of different aspects of the model. First, we look at the fluency model and compare the cluster smoothing (CS) approach with softmax temperature (TEMPx with x being the temperature) commonly used for generation in LM-integrated models (Chorowski and Jaitly, 2016) as well as no adjustment (NA). Second, we vary the 3-layer representation out of ELMo forward language model to do contextual matching (bot/mid/top: bottom/middle/top layer only, avg: average of 3 layers, cat: concatenation of all layers).

	abstractive			extractive	
Models	R1	R2	RL	F1	CR
CS + cat	26.48	10.05	24.41	60.90	0.38
CS + avg	26.34	9.79	24.23	60.09	0.38
CS + top	26.21	9.69	24.14	62.18	0.34
CS + mid	25.46	9.39	23.34	59.32	0.40
CS + bot	15.29	3.95	14.06	21.14	0.23
TEMP5 + cat	26.31	9.38	23.60	52.10	0.43
TEMP10 + cat	25.63	8.82	22.86	42.33	0.47
NA + cat	24.81	8.89	22.87	49.80	0.32

Table 3: Comparison of different model choices. The top section evaluates the effects of contextual representation in the matching model, and the bottom section evaluates the effects of different smoothing methods in the fluency model.

Results show the effectiveness of our cluster smoothing method for the vocabulary adaptive language model p_{fm}, although temperature smoothing is an option for abstractive datasets. Additionally Contextual embeddings have a huge impact on performance. When using word embeddings (bottom layer only from ELMo language model) in our contextual matching model p_{cm}, the summarization performance drops significantly to below simple baselines as demonstrated by score decrease. This is strong evidence that encoding independent tokens in a sequence with generic language model hidden states helps maintain the contextual flow. Experiments also show that even when only using p_{cm} (by setting $\lambda = 0$), utilizing the ELMo language model states allows the generated sequence to follow the source $\mathbf{x}$ closely, whereas normal context-free word embeddings would fail to do so.

Table 4 shows some examples of our unsupervised generation of summaries, compared with the human reference, an attention based seq-to-seq model we trained using all the Gigaword parallel data, and the GAN-based unsupervised system from Wang and Lee (2018). Besides our default of using all ELMo layers, we also show generations by using the top and bottom (context-independent)

I: japan 's nec corp. and UNK computer corp. of the united states said wednesday they had agreed to join forces in supercomputer sales
G: nec UNK in computer sales tie-up
s2s: nec UNK to join forces in supercomputer sales
GAN: nec corp. to join forces in sales
CM (cat): nec agrees to join forces in supercomputer sales
CM (top): nec agrees to join forces in computer sales
CM (bot): nec to join forces in supercomputer sales

I: turnout was heavy for parliamentary elections monday in trinidad and tobago after a month of intensive campaigning throughout the country , one of the most prosperous in the caribbean
G: trinidad and tobago poll draws heavy turnout by john babb
s2s: turnout heavy for parliamentary elections in trinidad and tobago
GAN: heavy turnout for parliamentary elections in trinidad
CM (cat): parliamentary elections monday in trinidad and tobago
CM (top): turnout is hefty for parliamentary elections in trinidad and tobago
CM (bot): trinidad and tobago most prosperous in the caribbean

I: a consortium led by us investment bank goldman sachs thursday increased its takeover offer of associated british ports holdings , the biggest port operator in britain , after being threatened with a possible rival bid
G: goldman sachs increases bid for ab ports
s2s: goldman sachs ups takeover offer of british ports
GAN: us investment bank increased takeover offer of british ports
CM (cat): us investment bank goldman sachs increases shareholdings
CM (top): investment bank goldman sachs increases investment in britain
CM (bot): britain being threatened with a possible bid

Table 4: Abstractive sentence summary examples on Gigaword test set. I is the input, G is the reference, s2s is a supervised attention based seq-to-seq model, GAN is the unsupervised system from Wang and Lee (2018), and CM is our unsupervised model. The third example is a failure case we picked where the sentence is fluent and makes sense but misses the point as a summary.

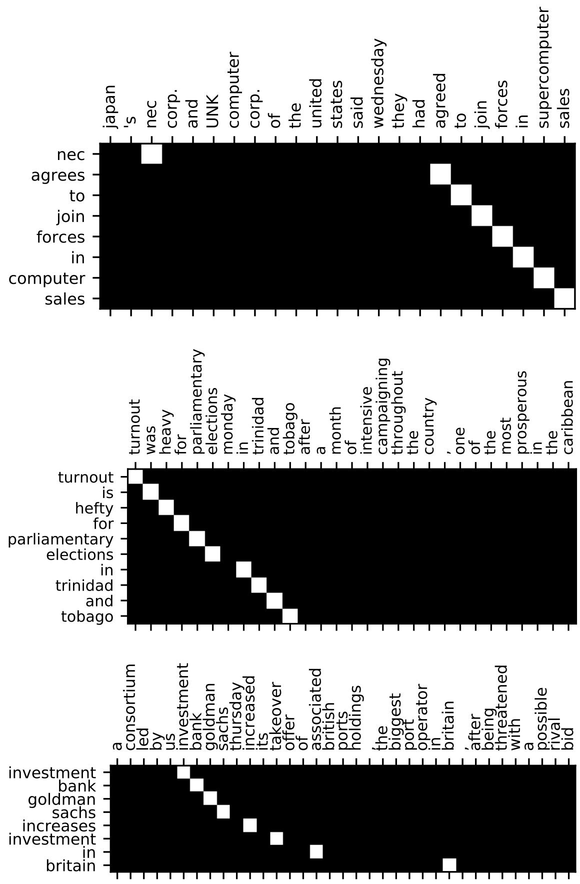

Figure 2: Examples of alignment results generated by our unsupervised method between the abstractive summaries and corresponding source sentences in the Gigaword test set.

layer only. Our generation has fairly good qualities, and it can correct verb tenses and paraphrase automatically. Note that top representation actually finds more abstractive summaries (such as in example 2), and the bottom representation fails to focus on the proper context. The failed examples are mostly due to missing the main point, as in example 3, or the summary needs to reorder tokens in the source sequence. Moreover, as a byproduct, our unsupervised method naturally generates hard alignments between summary and source sentences in the contextual matching process. We show some examples in Figure 2 corresponding to the sentences in Table 4.

5 Conclusion

We propose a novel methodology for unsupervised sentence summarization using contextual matching. Previous neural unsupervised works mostly adopt complex encoder-decoder frameworks. We achieve good generation qualities and competitive evaluation scores. We also demonstrate a new way of utilizing pre-trained generic language models for contextual matching in untrained generation. Future work could be comparing language models of different types and scales in this direction.

Acknowledgements

We would like to thank Yuntian Deng and Yoon Kim for useful discussions. This work was supported by NSF 1845664 and research awards from Google, Oracle, and Facebook.

References

Franz Aurenhammer. 1991. Voronoi diagramsa survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161.

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.

Jan Chorowski and Navdeep Jaitly. 2016. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*.

Thibault Fevry and Jason Phang. 2018. Unsupervised sentence compression using denoising autoencoders. *arXiv preprint arXiv:1809.02669*.

Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

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

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*.

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

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Emily Pitler. 2010. Methods for sentence compression.

Colin Raffel, Minh-Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck. 2017. Online and linear-time attention by enforcing monotonic alignments. *arXiv preprint arXiv:1704.00784*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Raphael Schumann. 2018. Unsupervised abstractive sentence summarization using length controlled variational autoencoder. *arXiv preprint arXiv:1809.05233*.

Yau-Shian Wang and Hung-Yi Lee. 2018. Learning to encode text as human-readable summaries using generative adversarial networks. *arXiv preprint arXiv:1810.02851*.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomas Kocisky. 2016a. The neural noisy channel. *arXiv preprint arXiv:1611.02554*.

Lei Yu, Jan Buys, and Phil Blunsom. 2016b. Online segment to segment neural transduction. *arXiv preprint arXiv:1609.08194*.

Yang Zhao, Zhiyuan Luo, and Akiko Aizawa. 2018. A language model based evaluator for sentence compression. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 170–175.