

A Study on Seq2seq for Sentence Compression in Vietnamese

Thi-Trang Nguyen^{1,2}, Huu-Hoang Nguyen², Kiem-Hieu Nguyen¹

¹Hanoi Uni. of Science and Technology, Hanoi, Vietnam

²VCCorp, Hanoi, Vietnam

{nguyenthtrang03,hoangnguyenhuu}@admicro.vn, hieunk@soict.hust.edu.vn

Abstract

Text summarization is an important yet challenging task in natural language processing. In this paper, we investigate Pointer Generator Networks for sentence compression. Using Vietnamese as a case study, our model could yield sentence summaries with high quality of syntax, factual correctness and completeness. Interestingly, we demonstrate that only a simple filtering technique is required to generate training data of sentence-summary pairs without any human annotation

1 Introduction

Text summarization is an important and challenging aspect of natural language processing. The ultimate goal is to generate summary that retains essential information of the original text. Extractive approaches attempts to identify salient parts of the source text and assembles them into a summary. In contrast, abstractive approaches uses language modelling technique which is conditioned on original text to generate summary that might have different sentence structure and novel words. This paper focuses on sentence compression, which could be used in both abstractive and extractive summarization systems or as a stand-alone application. Sentence compression methods could be broadly classified into two categories, deletion-based and abstractive models. Deletion-based approach depends upon efforts to find and delete unimportant words or phrases in the original sentence. A shorter sentence is then produced by stitching together remaining fragments. An abstractive sentence compressor,

on the other hand, is essentially similar to a document or multi-document summarizer except that it only takes the original sentence as context and it produces one sentence instead of a multi-sentence summary.

Recent success of sequence-to-sequence (seq2seq) framework in machine translation paves the way for the emergence of neural abstractive summarization [Chopra et al.2016, Wubben et al.2016, Nallapati et al.2016, See et al.2017, Rush et al.2015]. Although there have been several studies in Vietnamese text summarization [Nguyen and Nguyen2011, Dac et al.2017], we haven't been aware of any neural networks-based methods, partially due to the lack of a large volume of high-quality training data.

In a newswire article, we observe that *sapo*¹ is typically a longer version of title. Based on this observation, we could collect a large amount of sentence - summary pairs from online newswire. It then requires a simple filtering step based on textual similarity to remove unwanted title - *sapo* pairs. We rely on a particular model in seq2seq family, i.e. Pointer Generation Networks [See et al.2017] to learn summarizing from a large amount of training pairs.

As suggested in [Kryscinski et al.2019], we evaluate generated summaries using human judgement alongside ROUGE metric [Lin2004]. Despite inevitable noise in training data, our model could yield summaries with high quality of syntax, factual correctness and completeness.

¹Sapo is a paragraph, usually containing one sentence, that follows the title and precedes the first sentence of an article.

2 Related work

There are two main approaches to sentence compression: deletion-based and abstractive approach. In deletion-based approach, one has to decide whether to keep or remove each token in the original sentence [Jing2000, Clarke and Lapata2008, Fevry and Phang2018, Wang et al.2017, Galanis and Androutsopoulos2010]. An early work of Jing [Jing2000] applied linguistic rules and lexicon to parse tree to remove non-salient phrases from the original sentence. Recently, Filippova [Filippova et al.2015] adopted an LSTM network to resolve the problem as sequence labeling. Following this direction, Wang et al [Wang et al.2017] show that syntax could be useful for LSTM-based sentence compression.

Abstractive approach to sentence compression is more powerful in that it considers all the operations including deletion, reordering, substitution and insertion. Cohn and Lapata [Cohn and Lapata2008] learn a set of parse tree transduction rules from a training dataset of sentence-summary pairs. Recently, seq2seq framework has been studied for this task [Chopra et al.2016, Nallapati et al.2016, See et al.2017]. Seq2Seq realizes encoder-decoder paradigm where the encoder encodes an input sequence into hidden states, from which the decoder then generates the output sequence. Taken together, the attention mechanism automatically align input and output sequences, that boosts the performance of seq2Seq significantly.

Besides just that, there are researches in unsupervised sentence compression. In [Fevry and Phang2018], Fevry and Phang train a denoising auto-encoder to recover the original, constructing an end-to-end training regime. In [Baziotis et al.2019], Baziotis et al present a seq2seq autoencoder which consists of two chained encoder-decoder pairs. It learns to restore the original sentence while forcing the middle hidden sequence to generate important information in the sentence, thus generate its summary without parallel training data.

There are several studies on sentence compression in Vietnamese. Nguyen [Nguyen and Horiguchi2003] learns transition rules from example pairs to generate summaries in English and Vietnamese from an original English sentence. In another work, the author [Nguyen et al.2004] uti-

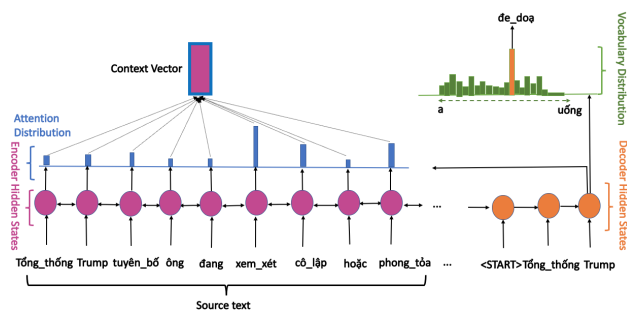


Figure 1: Vanilla Seq2seq with attention

lized Hidden Markov Models for deletion-based sentence compression. In [Nguyen and Nguyen2011], Nguyen et al applied Viterbi decoding to find the most likelihood substrings and then concatenate them to generate compression. In [Tran et al.2015], Tran proposed Conditional Random Fields using information on meaningful chunks as feature. In [Tran and Nguyen2018], Tran proposed a three-phase method for summarizing paragraph. It first builds a graph to represent input content with coreference resolution. The graph is then transformed into abstract semantic representation. New sentence is finally generated from this representation.

3 Seq2seq models

In this section, we briefly show (1) baseline Seq2seq model, (2) pointer-generator model, and (3) coverage mechanism that can be added to either of the first two models. The original paper contains far more details and in-depth specifications [See et al.2017].

3.1 Vanilla seq2seq

A vanilla seq2seq framework for abstractive summarization is composed of an encoder and a decoder with attention mechanism. The encoder is a single-layer bidirectional LSTM. In an attention-based encoder-decoder architecture (shown in Figure 1), the decoder (a single-layer unidirectional LSTM) not only takes the encoded representations of the source sequence as input, but also selectively focuses on parts of the sequence at each decoding step. On the other hand, that tells the decoder where to look up to produce the next word. The attention distribution is calculated as in below:

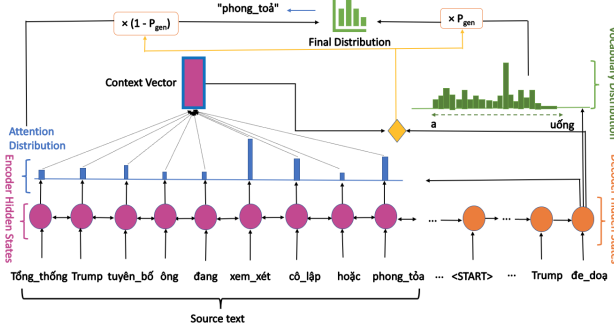


Figure 2: Pointer-generator networks.

$$e_i^t = \nu^T \tanh(\mathbf{W}_h \mathbf{h}_i + \mathbf{W}_s \mathbf{s}_t + b_{attn}) \quad (1)$$

$$\mathbf{a}^t = \text{softmax}(e^t) \quad (2)$$

where \mathbf{h}_i is encoder hidden state after feeding the token w_i to the encoder network, \mathbf{s}_t is generated by the decoder when receiving the previous word representation at each step t . ν , \mathbf{W}_h , \mathbf{W}_s and b_{attn} are learnable parameters.

\mathbf{P}_{vocab} , probability distribution over all words in the decoder vocabulary, is produced by concatenating the context vector \mathbf{h}_i^* with the decoder state \mathbf{s}_t and feeding through two linear layers. Details are described in the following formulas:

$$\mathbf{h}_i^* = \sum_i \mathbf{a}_i^t \mathbf{h}_i \quad (3)$$

$$\mathbf{P}_{vocab} = \text{softmax}(\mathbf{V}'(\mathbf{V}[\mathbf{s}_t, \mathbf{h}_i^*] + \mathbf{b}) + \mathbf{b}') \quad (4)$$

where \mathbf{V} , \mathbf{V}' , \mathbf{b} and \mathbf{b}' are learnable parameters.

The overall loss for whole sequence is shown below:

$$\text{loss} = \frac{1}{T} \sum_{t=0}^T -\log \mathbf{P}_{vocab}(\mathbf{w}_t^*) \quad (5)$$

where the loss for each timestep t is the negative log likelihood of the target word \mathbf{w}_t^* for that time step

3.2 Pointer-generator network

Because the baseline model are restricted to their pre-set vocabulary, the ability to generate out-of-vocabulary words (OOV) is one of the primary advantages of pointer-generator models. The pointer-generator network allows the model to generate tokens by copying from the input sequence. The architecture of pointer-generator is described on figure 2. It is equipped with a “soft-switch”, decoder vocabulary or point to one in the source article at each decoding step. The soft-switch is explicitly modeled by

$$p_{gen} = \sigma(\mathbf{w}_{h^*}^T \mathbf{h}_i^* + \mathbf{w}_s^T \mathbf{s}_t + \mathbf{w}_x^T x_t + b_{ptr}) \quad (6)$$

where vectors \mathbf{w}_{h^*} , \mathbf{w}_s , \mathbf{w}_x and scalar b_{ptr} are learnable parameter and σ is the sigmoid function. For each sequence let the *extended vocabulary* denote the union of the vocabulary, and all words appearing in the source sequence. The vocabulary distribution over an extended vocabulary is calculated by:

$$\mathbf{P}(w) = p_{gen} \mathbf{P}_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} \mathbf{a}_i^t \quad (7)$$

Note that if w is OOV, $\mathbf{P}_{vocab}(w)$ is zero; if w does not appear in the source sequence, then $\sum_{i:w_i=w} \mathbf{a}_i^t$ is zero. The loss function is described similarly on the seq2seq attention models as equations (5) and (6), but with respect to probability distribution $\mathbf{P}(w)$ given in equation (8).

3.3 Coverage mechanism

Coverage model was created to solve repetition problem for seq2seq models. In this model, they first defined a *coverage vector* \mathbf{c}^t as the sum of attention distributions of the previous decoding steps:

$$\mathbf{c}^t = \sum_{t'=0}^{t-1} \mathbf{a}^{t'} \quad (8)$$

Thus, it contains the accumulated attention information on each token in the source sequence during the previous decoding steps. Note that \mathbf{c}^0 is zero vector, because none of the source sequence has been covered on the first timestep. The coverage vector

will then be used as an extra input to the attention mechanism, changing (1) to:

$$e_i^t = \nu^T \tanh(\mathbf{W}_h \mathbf{h}_i + \mathbf{W}_s \mathbf{s}_t + \mathbf{w}_c \mathbf{c}_i^t + b_{attn}) \quad (9)$$

where \mathbf{w}_c is a learnable parameter vector of same length as ν

Finally, the *coverage loss* $covloss_t$ is reweighted by some hyperparameters λ , is added to primary loss function (5) to yield a new loss function:

$$covloss_t = \sum_i \min(\mathbf{a}_i^t, \mathbf{c}_i^t) \quad (10)$$

$$loss_t = -\log P(\mathbf{w}_t^*) + \lambda \sum_i \min(\mathbf{a}_i^t, \mathbf{c}_i^t) \quad (11)$$

4 Experiments

In all experiments, we used pytorch implementation of [See et al.2017]².

4.1 Data preparation

Manually building a large dataset really requires a lot of commitment from annotators and a great deal of time and effort. This would appear to be infeasible. In this section, we explain how we obtained a corpus of sentences and their compressions.

The underlying idea is to harvest news articles from the Internet where the *sapo* appears to be an extension of the title of an article, and vice versa, the title appears to be a compression of the *sapo*. Using a news crawler, we collected a large number of news items in Vietnamese. Word segmentation was processed by UETSegmenter³. From every article, we examined the correlation between *sapo* (S) and title (T). For each pair of (S, T), we used some filters and a simple scoring function to decide which pairs should be retained:

- The number of words in S must be greater than 25 (excluding punctuation)
- The number of words in T is within (8, 15] (excluding punctuation)

²https://github.com/atulkum/pointer_summarizer

³<https://github.com/phongnt570/UETsegmenter>

- The word duplicate rate - $d(S, T)$ is greater than a threshold α .

We obtained 1M from 3M pairs of (S, T) with $\alpha = 0.25$. We use 900K pairs as training set and 100K pairs as development set. To test our model, we asked annotators with good language skill and good knowledge in news summarization to manually select from a large candidate pool the pairs in which title is indeed a summary of *sapo* sentence. The test-set, namely PegaTest, contains 9K of such sentence pairs.

4.2 Hyper-parameters

In this subsection, we present hyper-parameters which were tuned on the development set. For all experiments, our models have 256-dimensional hidden states and 128-dimensional word embeddings. We used Adagrad optimizer with mini-batch size 32. Gradient clipping with a maximum gradient norm of 2 was used, but we didn't use any form of regularization. We chose an learning rate of $\eta_0 = 0.15$ and an initial accumulator value of 0.1 when training *vanilla seq2seq* and *Pointer* models. Learning rate was reduced to 0.1 on *Pointer+coverage* model.

When it comes to the *Pointer+coverage* model, we experimented in two versions. In the first version, we followed [See et al.2017] to first learn a pointer-generator network and then add the coverage mechanism into loss function and continue to train for 40000 steps. In the second version, we simply trained with coverage from the first iteration with the weight of coverage loss $\lambda = 0.5$.

In all our models, we used a vocabulary of 50k words for both source and summary sentences. During training and test time, we truncated source sentence to 50 tokens. In decoding, the length of the summary were limited to 25 tokens. On the other hand, our summaries were produced using beam search with beam size 4. We also used early stopping based on development set in which convergence is reached after around 5 epochs.

4.3 Evaluation on Rouge

Rouge metric is used for evaluation using PegaTest. In this experiment, we compare the following models:

Table 1: Evaluation on ROUGE

Model	ROUGE-1			ROUGE-2			ROUGE-L		
	R	P	F	R	P	F	R	P	F
Vanilla seq2seq (no filter)	52.25	64.23	56.71	32.96	40.16	35.60	43.42	53.16	47.03
Vanilla seq2seq	69.29	63.63	65.59	49.34	45.51	46.77	58.97	54.33	55.91
Pointer	69.80	65.69	66.91	50.28	47.38	48.23	59.43	55.98	57.00
Pointer+coverage	65.23	69.06	66.36	46.70	49.27	47.38	55.16	58.25	56.00
Pointer+coverage [See et al.2017]	67.09	69.65	67.65	48.83	50.50	49.11	57.21	59.25	57.62

Table 2: Evaluation by humans

Model	Syntax	Factual correctness	Completeness
Vanilla seq2seq	84.67	60.62	68.70
Pointer	91.00	79.48	75.89
Pointer+Coverage	91.30	80.29	80.77
Pointer+Coverage [See et al.2017]	92.00	75.00	73.96

- *Vanilla seq2seq*: Encoder-decoder with attention mechanism.
- *Vanilla seq2seq (no filter)*: We want to investigate the effect of filtering noisy data as described in Section 4.1.
- *Pointer*: Pointer-generator network.
- *Pointer+coverage*: Pointer-generator network with coverage mechanism.

Firstly, as shown in Table 1, it is clear that filtering noisy data gains a substantial enhancement in quality of summaries. In our experiments, we noticed that, in terms of efficiency, not only training data was reduced, but the models also converged faster than when training on full data.

Secondly, pointer-network and the coverage mechanism bring a significant improvement over vanilla seq2seq model. As already mentioned in their paper, our version of Pointer-network resulted in a slightly worse Rouge score than the original version in [See et al.2017].

As pointed out in [Kryscinski et al.2019] and several works, Rouge metric is insufficient for evaluating summarization. In our experiments, we decided to further assess summary quality by human (Section 4.4).

4.4 Human evaluation

We evaluated summary on three criteria, namely syntax, factual correctness and completeness. We randomly selected 300 sapos and evaluated summaries generated by our models. The evaluation will comply with the following rule: Firstly, a summary that has correct syntax will be further considered for factual correctness. A summary that describes a correct fact as in the original sentence will then be taken into account for completeness. Syntax and factual correctness will have score as 0 (false) or 1 (true). Score of completeness is in the range [0,10], which shows the amount of important information preserved, in comparison with golden summaries.

Table 2 shows the evaluation results. The percentage of correct syntax sentences of vanilla seq2seq model is appreciably lower than the other two models. It’s just in as few as 84.67% of samples. It can be observed that the model seq2seq with only attention, unsuccessfully learned Vietnamese grammar.

Pointer is on par with Pointer+coverage in factual correctness. Pointer+coverage [See et al.2017] lags behind with 5% below. Pointer+coverage performs the best on completeness. Two conclusions could be drawn from these results: Coverage mechanism proves to be useful for generating high quality compressions; and most importantly, ROUGE met-

<p>1. (S): Ngày 15/8 , Công_ty VinSmart đã có phản_hồi với báo_giới về video so_sánh thiết_kế và linh_kiện smartphone Vsmart_Live với model Meizu 16X tới từ Trung_Quốc .</p> <p>(T): VinSmart nói gì về video so_sánh điện_thoại Vsmart_Live với Meizu</p> <p>(Vanilla ses2seq): [UNK] [UNK] - Tin_tức mới nhất về thiết_kế và linh_kiện smartphone [UNK]</p> <p>(Pointer): Công_ty VinSmart phản_hồi về video so_sánh thiết_kế và linh_kiện smartphone</p> <p>(Pointer+coverage): VinSmart phản_hồi về video so_sánh thiết_kế và linh_kiện smartphone Vsmart_Live</p>
<p>2. (S): Bệnh_viện Dã_chiến cấp 2 số 2 được điều_chuyển nguyên_trạng từ Học_viện Quân_y về Cục Giữ_hoà_bình Việt_Nam , bao_gồm chức_năng , nhiệm_vụ , tổ_chức , biên_chế quân_số , trang_thiết_bị .</p> <p>(Vanilla ses2seq): Bệnh_viện Dã_chiến cấp 2 cấp 2 số 2 được điều_chuyển</p> <p>(Pointer): Bệnh_viện Dã_chiến cấp 2 được điều_chuyển nguyên_trạng từ Học_viện Quân_y</p> <p>(Pointer+coverage): Bệnh_viện Dã_chiến cấp 2 được điều_chuyển nguyên_trạng từ Học_viện Quân_y</p>
<p>3. (S): Do Trung_Quốc gây khó_đễ và kiểm_soát chặt cửa_khẩu , hàng_loạt trái_cây Việt_Nam như : dưa_hầu , dưa_xiêm và thanh_long ... đều mất_giá từ 50 % .</p> <p>(T): Hàng_loạt trái_cây Việt mất_giá vì Trung_Quốc siết đầu_vào .</p> <p>(Vanilla ses2seq): Hàng_loạt trái_cây Việt_Nam ‘ bốc_hơi ’ từ 50 %</p> <p>(Pointer): hàng_loạt trái_cây Việt_Nam mất_giá từ 50 % từ 50 %</p> <p>(Pointer+coverage): hàng_loạt trái_cây Việt_Nam đều mất_giá từ 50 %</p>
<p>4. (S): Một nguồn_tin từ Chính_phủ Nhật_Bản ngày 16/8 cho_biết , nước này vừa đề_xuất với Mỹ rằng Tokyo sẵn_sàng cung_cấp người_máy_tự_động sử_dụng cho việc dỡ bỏ các cơ_sở hạt_nhân của Triều_Tiên .</p> <p>(T): Nhật_Bản đề_xuất dùng robot dỡ bỏ các cơ_sở hạt_nhân của Triều_Tiên .</p> <p>(Vanilla ses2seq): Nhật_Bản đề_xuất cho Mỹ dỡ bỏ cơ_sở hạt_nhân Triều_Tiên</p> <p>(Pointer): Nhật_Bản sẵn_sàng cung_cấp người_máy_tự_động sử_dụng cho Triều_Tiên .</p> <p>(Pointer+coverage): Nhật_Bản đề_xuất dỡ bỏ các cơ_sở hạt_nhân của Triều_Tiên.</p>
<p>5. (S): Virgil van Dijk đã xuất_sắc vượt_qua Leo_Messi và Cristiano_Ronaldo để giành giải_thưởng “ Cầu_thủ xuất_sắc nhất mùa_giải 2018/2019 ” của UEFA với số điếm 305 .</p> <p>(T): Virgil van Dijk giành danh_hiệu “ Cầu_thủ xuất_sắc nhất mùa_giải 2018/2019 ” của UEFA .</p> <p>(Vanilla ses2seq): [UNK] giành giải Cầu_thủ xuất_sắc nhất mùa_giải 2018/2019</p> <p>(Pointer): Cristiano_Ronaldo giành giải_thưởng “ Cầu_thủ xuất_sắc nhất mùa_giải 2018/2019 ” .</p> <p>(Pointer+coverage): Cristiano_Ronaldo giành giải Cầu_thủ xuất_sắc nhất mùa_giải 2018/2019 .</p>

Figure 3: Example of compresses sentences.

ric doesn’t always correlate with human evaluation.

Furthermore, we provide a detailed analysis of grammatical and semantic issues of seq2seq models through the examples in figure 3.

Vanilla seq2seq usually generates [UNK] tokens or uses novel words that change the meaning of the original sentence. Proper names and numbers are usually replaced by [UNK]. In Example 3, the verb ‘mất_giá’ (devalue) was replaced by the verb ‘bốc_hơi’ (evaporate) making the summary sentence semantically incorrect. Even more catastrophically, the condensed sentences sometimes evolve into repetitive nonsense, such as the second sample.

The other two models show a significant improvement when considering these issues. The verb ‘mất_giá’ (devalue) was captured exactly in Pointer but the repetition problem has not yet been resolved. When the coverage mechanism is added, accurate condensed information is generated. Moreover, both proper names ‘VinSmart’ and ‘Vsmart_live’ in Example 1 have been correctly presented by the last model.

In Example 4, all three models generate sentences with correct grammar but none of them is capable of retaining the main idea of the original sentence. In Example 5, vanilla seq2seq unsuccessfully deals with OOV words (‘Virgil van Dijk’). Meanwhile, both Pointer and Pointer+coverage select ‘Cristiano_Ronaldo’ as misleading subject of the sentence.

The models also tend to learn writing style of news title such as using a location with ‘:’, or subject ellipsis. This suggests that further filtering could be applied to enhance data quality.

5 Conclusions and Future Work

In this work, we have presented Pointer-Generator Networks for Vietnamese sentence compression. Results are significant given no requirement on manual annotation. However, the generated summaries are far from perfect.

As a next step, we would like to further improve the summary quality, in a data-driven way, as well as scale this system to generate paragraph-level summaries. Both pose additional challenges in terms of efficient alignment and consistency in generation. Another direction is delving into seq2seq

framework, including recent transformer-based alternative, to increase model interpretability.

References

- Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQ³: Differentiable sequence-to-sequence-to-sequence auto-encoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 673–681, Minneapolis, Minnesota, June. Association for Computational Linguistics.
- 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, San Diego, California, June. Association for Computational Linguistics.
- James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *J. Artif. Intell. Res. (JAIR)*, 31:399–429, 01.
- Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK, August. Coling 2008 Organizing Committee.
- Viet Lai Dac, Truong Son Nguyen, and Le Minh Nguyen. 2017. Deletion-based sentence compression using bi-enc-dec LSTM. In *Computational Linguistics - 15th International Conference of the Pacific Association for Computational Linguistics, PACLING 2017, Yangon, Myanmar, August 16-18, 2017, Revised Selected Papers*, pages 249–260.
- Thibault Fevry and Jason Phang. 2018. Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422, Brussels, Belgium, October. Association for Computational Linguistics.
- 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, Lisbon, Portugal, September. Association for Computational Linguistics.
- Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 885–893, Los Angeles, California, June. Association for Computational Linguistics.
- Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *In Proceedings of the 6th Applied Natural Language Processing Conference*, pages 310–315.
- Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China, November. Association for Computational Linguistics.
- Chin Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August. Association for Computational Linguistics.
- Le Minh Nguyen and Susumu Horiguchi. 2003. A sentence reduction using syntax control. In *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*, pages 146–152, Sapporo, Japan, July. Association for Computational Linguistics.
- Thi Thu Ha Nguyen and Huu Quynh Nguyen. 2011. Concatenate the most likelihood substring for generating vietnamese sentence reduction. *International journal of engineering and technology*, 3:203–207.
- Le Minh Nguyen, Susumu Horiguchi, Akira Shimazu, and Tu Bao Ho. 2004. Example-based sentence reduction using the hidden markov model. In *ACM Transactions on Asian Language Information Processing (TALIP)*.
- Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September. Association for Computational Linguistics.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-

- generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July. Association for Computational Linguistics.
- Trung Tran and Dang Tuan Nguyen. 2018. Text generation from abstract semantic representation for summarizing vietnamese paragraphs having co-references. *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 93–98.
- Nhi Thao Tran, Van Giau Ung, An Vinh Luong, Minh Quoc Nghiem, and Luu Thuy Ngan Nguyen. 2015. Improving vietnamese sentence compression by segmenting meaning chunks. *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)*, pages 320–323.
- Liangguo Wang, Jing Jiang, Hai Leong Chieu, Chen Hui Ong, Dandan Song, and Lejian Liao. 2017. Can syntax help? improving an LSTM-based sentence compression model for new domains. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1385–1393, Vancouver, Canada, July. Association for Computational Linguistics.
- Sander Wubben, Emiel Kraemer, Antal van den Bosch, and Suzan Verberne. 2016. Abstractive compression of captions with attentive recurrent neural networks. In *Proceedings of the 9th International Natural Language Generation conference*, pages 41–50, Edinburgh, UK, September 5-8. Association for Computational Linguistics.