A Cascade Approach to Neural Abstractive Summarization with Content Selection and Fusion

Logan Lebanoff,[♠] **Franck Dernoncourt,**[♢] **Doo Soon Kim,**[♢] **Walter Chang,**[♢] **Fei Liu**[♠]
[♠]Computer Science Department, University of Central Florida, Orlando, FL
[♢]Adobe Research, San Jose, CA
loganlebanoff@knights.ucf.edu {dernonco,dkim,wachang}@adobe.com
feiliu@cs.ucf.edu

Abstract

We present an empirical study in favor of a cascade architecture to neural text summarization. Summarization practices vary widely but few other than news summarization can provide a sufficient amount of training data enough to meet the requirement of end-to-end neural abstractive systems which perform content selection and surface realization jointly to generate abstracts. Such systems also pose a challenge to summarization evaluation, as they force content selection to be evaluated along with text generation, yet evaluation of the latter remains an unsolved problem. In this paper, we present empirical results showing that the performance of a cascaded pipeline that separately identifies important content pieces and stitches them together into a coherent text is comparable to or outranks that of end-to-end systems, whereas a pipeline architecture allows for flexible content selection. We finally discuss how we can take advantage of a cascaded pipeline in neural text summarization and shed light on important directions for future research.

1 Introduction

There is a variety of successful summarization applications but few can afford to have a large number of annotated examples that are sufficient to meet the requirement of end-to-end neural abstractive summarization. Examples range from summarizing radiology reports (Jing et al., 2019; Zhang et al., 2020) to congressional bills (Kornilova and Eidelman, 2019) and meeting conversations (Mehdad et al., 2013; Li et al., 2019; Koay et al., 2020). The lack of annotated resources suggests that end-to-end systems may not be a "one-size-fits-all" solution to neural text summarization. There is an increasing need to develop cascaded architectures to allow for customized content selectors to be combined with general-purpose neural text generators to realize the full potential of neural abstractive summarization.

We advocate for explicit content selection as it allows for a rigorous evaluation and visualization of intermediate results of such a module, rather than associating it with text generation. Existing neural abstractive systems can perform content selection implicitly using end-to-end models (See et al., 2017; Celikyilmaz et al., 2018; Raffel et al., 2019; Lewis et al., 2020), or more explicitly, with an external module to select important sentences or words to aid generation (Tan et al., 2017; Gehrmann et al., 2018; Chen and Bansal, 2018; Kryściński et al., 2018; Hsu et al., 2018; Lebanoff et al., 2018, 2019b; Liu and Lapata, 2019). However, content selection concerns not only the selection of important segments from a document, but also the cohesiveness of selected segments and the amount of text to be selected in order for a neural text generator to produce a summary.

In this paper, we aim to investigate the feasibility of a cascade approach to neural text summarization. We explore a constrained summarization task, where an abstract is created one sentence at a time through a cascaded pipeline. Our pipeline architecture chooses one or two sentences from the source document, then highlights their summary-worthy segments and uses those as a basis for composing a summary sentence. When a pair of sentences are selected, it is important to ensure that they are *fusible*—there exists cohesive devices that tie the two sentences together into a coherent text—to avoid generating nonsensical outputs (Geva et al., 2019; Lebanoff et al., 2020). Highlighting sentence segments allows us to perform fine-grained content selection that guides the neural text generator to stitch selected segments into a coherent sentence. The contributions of this work are summarized as follows.

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 529–535

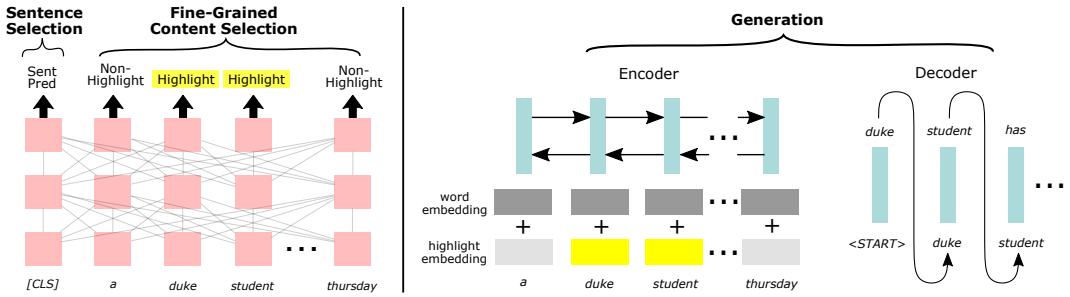

Figure 1: Model architecture. We divide the task between two main components: the first component performs sentence selection and fine-grained content selection, which are posed as a classification problem and a sequence-tagging problem, respectively. The second component receives the first component's outputs as supplementary information to generate the summary. A cascade architecture provides the necessary flexibility to separate content selection from surface realization in abstractive summarization.

- We present an empirical study in favor of a cascade architecture for neural text summarization. Our cascaded pipeline chooses one or two sentences from the document and highlights their important segments; these segments are passed to a neural generator to produce a summary sentence.
- Our quantitative results show that the performance of a cascaded pipeline is comparable to or outranks that of end-to-end systems, with added benefit of flexible content selection. We discuss how we can take advantage of a cascade architecture and shed light on important directions for future research.[1]

2 A Cascade Approach

Our cascaded summarization approach focuses on shallow abstraction. It makes use of text transformations such as sentence shortening, paraphrasing and fusion (Jing and McKeown, 2000) and is in contrast to deep abstraction, where a full semantic analysis of the document is often required. A shallow approach helps produce abstracts that convey important information while, crucially, remaining faithful to the original. In what follows, we describe our approach to select single sentences and sentence pairs from the document, highlight summary-worthy segments and perform summary generation conditioned on highlights.

Selection of Singletons and Pairs Our approach iteratively selects one or two sentences from the input document; they serve as the basis for composing a single summary sentence. Previous research suggests that 60-85% of human-written summary sentences are created by shortening a single sentence or merging a pair of sentences (Lebanoff et al., 2019b). We adopt this setting and present a coarse-to-fine strategy for content selection. Our strategy begins with selecting sentence singletons and pairs, followed by highlighting important segments of the sentences. Importantly, the strategy allows us to control which segments will be combined into a summary sentence—"compatible" segments come from either a single document sentence or a pair of *fusible* sentences. In contrast, when all important segments of the document are provided to a neural generator all at once (Gehrmann et al., 2018), it can happen that the generator arbitrarily stitches together text segments from unrelated sentences, yielding a summary that contains hallucinated content and fails to retain the meaning of the original document (Falke et al., 2019; Lebanoff et al., 2019a; Kryscinski et al., 2019).

We expect a sentence singleton or pair to be selected from the document if it contains salient content. Moreover, a pair of sentences should contain content that is compatible with each other. Given a sentence or pair of sentences from the document, our model predicts whether it is a valid instance to be compressed or merged to form a summary sentence. We follow (Lebanoff et al., 2019b) to use BERT (Devlin et al., 2019) to perform the classification. BERT is a natural choice since it takes one or two sentences and generates a classification prediction. It treats an input singleton or pair of sentences as a sequence of tokens. The tokens are fed to a series of Transformer block layers, consisting of multi-head self-attention modules. The first Transformer layer creates a contextual representation for each token, and each successive layer further refines those representations. An additional

[1]Our code is publicly available at https://github.com/ucfnlp/cascaded-summ

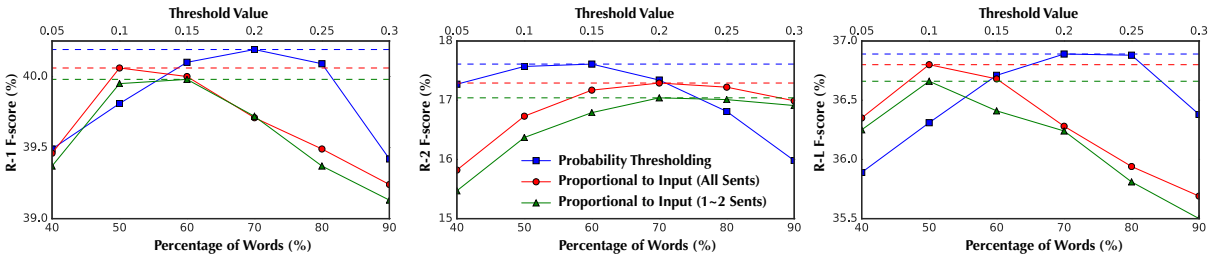

Figure 2: Comparison of various highlighting strategies. Thresholding obtains the best performance.

[CLS] token is added to contain the sentence representation. BERT is fine-tuned for our task by adding an output layer on top of the final layer representation $\mathbf{h}^L_{[\text{CLS}]}$ for sequence s, as seen in Eq. (1).

$$p_{\text{sent}}(s) = \sigma(\mathbf{u}^\top \mathbf{h}^L_{[\text{CLS}]}) \quad (1)$$

where $\mathbf{u}$ is a vector of weights and σ is the sigmoid function. The model predicts p_{sent} – whether the sentence singleton or pair is an appropriate one based on the [CLS] token representation. We describe the training data for this task in §3.

Fine-Grained Content Selection It is interesting to note that the previous architecture can be naturally extended to perform fine-grained content selection by highlighting important words of sentences. When two sentences are selected to generate a fusion sentence, it is desirable to identify segments of text from these sentences that are potentially compatible with each other. The coarse-to-fine method allows us to examine the intermediate results and compare them with ground-truth. Concretely, we add a classification layer to the final layer representation $\mathbf{h}^L_i$ for each token w_i (Eq. (2)). The per-target-word loss is then interpolated with instance prediction (one or two sentences) loss using a coefficient λ. Such a multi-task learning objective has been shown to improve performance on a number of tasks (Guo et al., 2019).

$$p_{\text{highlight}}(w_i) = \sigma(\mathbf{v}^\top \mathbf{h}^L_i) \quad (2)$$

where $\mathbf{v}$ is a vector of weights and σ is the sigmoid function. The model predicts $p_{\text{highlight}}$ for each token – whether the token should be included in the output fusion, calculated based on the given token's representation.

Information Fusion Given one or two sentences taken from a document and their fine-grained highlights, we proceed by describing a fusion process that generates a summary sentence from the selected content. Our model employs an encoder-decoder architecture based on pointer-generator networks that has shown strong performance on its own and with adaptations (See et al., 2017; Gehrmann et al., 2018). We feed the sentence singleton or pair to the encoder along with highlights derived by the fine-grained content selector, the latter come in the form of binary tags. The tags are transformed to a "*highlight-on*" embedding for each token if it is chosen by the content selector, and a "*highlight-off*" embedding for each token not chosen. The highlight-on/off embeddings are added to token embeddings in an element-wise manner; both highlight and token embeddings are learned. An illustration is shown in Figure 1.

Highlights provide a valuable intermediate representation suitable for shallow abstraction. Our approach thus provides an alternative to methods that use more sophisticated representations such as syntactic/semantic graphs (Filippova and Strube, 2008; Banarescu et al., 2013; Liu et al., 2015). It is more straightforward to incorporate highlights into an encoder-decoder fusion model, and obtaining highlights through sequence tagging can be potentially adapted to new domains.

3 Experimental Results

Data and Annotation To enable direct comparison with end-to-end systems, we conduct experiments on the widely used CNN/DM dataset (See et al., 2017) to report results of our cascade approach. We use the procedure described in Lebanoff et al. (2019b) to create training instances for the sentence selector and fine-grained content selector. Our training data contains 1,053,993 instances; every instance contains one or two candidate sentences. It is a positive instance if a ground-truth summary sentence can be formed by compressing or merging sentences of the instance, negative otherwise. For positive instances, we highlight all lemmatized unigrams appearing in the summary, excluding punctuation. We further add smoothing to the labels by highlighting single words that con-

System	R-1	R-2	R-L
SumBasic (Vanderwende et al., 2007)	34.11	11.13	31.14
LexRank (Erkan and Radev, 2004)	35.34	13.31	31.93
Pointer-Generator (See et al., 2017)	39.53	17.28	36.38
FastAbsSum (Chen and Bansal, 2018)	40.88	17.80	**38.54**
BERT-Extr (Lebanoff et al., 2019b)	41.13	**18.68**	37.75
BottomUp (Gehrmann et al., 2018)	**41.22**	**18.68**	38.34
BERT-Abs (Lebanoff et al., 2019b)	37.15	15.22	34.60
Cascade-Fusion (Ours)	40.10	17.61	**36.71**
Cascade-Tag (Ours)	**40.24**	**18.33**	36.14
GT-Sent + Sys-Tag	50.40	27.74	46.25
GT-Sent + Sys-Tag + Fusion	51.33	28.08	47.50
GT-Sent + GT-Tag	**74.80**	48.21	**67.40**
GT-Sent + GT-Tag + Fusion	72.70	**48.33**	67.06

(SYSTEM SENTS) A Duke student has admitted to hanging a noose made of rope from a tree near a student union, university officials said Thursday. The student was identified during an investigation by campus police and the office of student affairs and admitted to placing the noose on the tree early Wednesday, the university said.

(CASCADE-FUSION) A Duke student was identified during an investigation by campus police and the office of student affairs and admitted to placing the noose on the tree early Wednesday.

(GT SENTS) In a news release, it said the student was no longer on campus and will face student conduct review. Duke University is a private college with about 15,000 students in Durham, North Carolina.

(GT SENTS + FUSION) Duke University student was no longer on campus and will face student conduct review.

(REFERENCE) Student is no longer on Duke University campus and will face disciplinary review.

Table 1: (LEFT) Summarization results on CNN/DM test set. Our cascade approach performs comparable to strong extractive and abstractive baselines; oracle models using ground-truth sentences and segment highlights perform the best. (RIGHT) Example source sentences and their fusions. Dark highlighting is content taken from the first sentence, and light highlighting comes from the second. Our *Cascade-Fusion* approach effectively performs entity replacement by replacing "student" in the second sentence with "a Duke student" from the first sentence.

nect two highlighted phrases and by dehighlighting isolated stopwords. At test time, four highest-scored instances are selected per document; their important segments are highlighted by content selector then passed to the fusion step to produce a summary sentence each. The hyperparameter λ for weighing the per-target-word loss is set to 0.2 and highlighting threshold value is 0.15. The model hyperparameters are tuned on the validation split.

Summarization Results We show experimental results on the standard test set and evaluated by ROUGE metrics (Lin, 2004) in Table 1. The performance of our cascade approaches, *Cascade-Fusion* and *Cascade-Tag*, is comparable to or outranks a number of extractive and abstractive baselines. Particularly, *Cascade-Tag* does not use a fusion step (§2) and is the output of fine-grained content selection. *Cascade-Fusion* provides a direct comparison against BERT-Abs (Lebanoff et al., 2019b) that uses sentence selection and fusion but lacks a fine-grained content selector.

Our results suggest that a coarse-to-fine content selection strategy remains necessary to guide the fusion model to produce informative sentences. We observe that the addition of the fusion model has only a moderate impact on ROUGE scores, but the fusion process can reorder text segments to create true and grammatical sentences, as shown in Table 1. We analyze the performance of a number of oracle models that use ground-truth sentence selection (GT-Sent) and tagging (GT-Tag). When given ground-truth sentences as input, our cascade models achieve ∼10 points of improvement in all ROUGE metrics. When the models are also given ground-truth highlights, they achieve an additional 20 points of improvement. In a preliminary examination, we observe that not all highlights are included in the summary during fusion, indicating there is space for improvement. These results show that cascade architectures have great potential to generate shallow abstracts and future emphasis may be placed on accurate content selection.

How much should we highlight? It is important to quantify the amount of highlighting required for generating a summary sentence. Highlighting too much or too little can be unhelpful. We experiment with three methods to determine the appropriate amount of words to highlight. *Probability Thresholding* chooses a set threshold whereby all words that have a probability higher than the threshold are highlighted. When *Proportional to Input* is used, the highest probability words are iteratively highlighted until a target rate is reached. The amount of highlighting can be proportional to the total number of words per instance (one or two sentences) or per document, containing all sentences selected for the document.

We investigate the effect of varying the amount of highlighting in Figure 2. Among the three methods, probability thresholding performs the best, as it gives more freedom to content selection. If the model scores all of the words in sentences highly, then we should correspondingly highlight all of the words. If only very few words score highly, then

we should only pick those few.

Highlighting a certain percentage of words tend to perform less well. On our dataset, a threshold value of 0.15–0.20 produces the best ROUGE scores. Interestingly, these thresholds end up highlighting 58–78% of the words of each sentence. Compared to what the generator was trained on, which had a median of 31% of each sentence highlighted, the system's rate of highlighting is higher. If the model's highlighting rate is set to be similar to that of the ground-truth, it yields much lower ROUGE scores (cf. threshold value of 0.3 in Figure 2). This observation suggests that the amount of highlighting can be related to the effectiveness of content selector and it may be better to highlight more than less.

4 Conclusion

We present a cascade approach to neural abstractive summarization that separates content selection from surface realization. Importantly, our approach makes use of text highlights as intermediate representation; they are derived from one or two sentences using a coarse-to-fine content selection strategy, then passed to a neural text generator to compose a summary sentence. A successful cascade approach is expected to accurately select sentences and highlight an appropriate amount of text, both can be customized for domain-specific tasks.

Acknowledgments

We are grateful to the anonymous reviewers for their comments and suggestions. This research was supported in part by the National Science Foundation grant IIS-1909603.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 177–185, Honolulu, Hawaii. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Mor Geva, Eric Malmi, Idan Szpektor, and Jonathan Berant. 2019. DiscoFuse: A large-scale dataset for discourse-based sentence fusion. 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 3443–3455, Minneapolis, Minnesota. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2019. AutoSeM: Automatic task selection and mixing in multi-task learning. 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 3520–3531, Minneapolis, Minnesota. Association for Computational Linguistics.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141, Melbourne, Australia. Association for Computational Linguistics.

Baoyu Jing, Zeya Wang, and Eric Xing. 2019. Show, describe and conclude: On exploiting the structure information of chest x-ray reports. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6570–6580, Florence, Italy. Association for Computational Linguistics.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and paste based text summarization. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Jia Jin Koay, Alexander Roustai, Xiaojin Dai, Alec Dillon, and Fei Liu. 2020. How domain terminology affects meeting summarization performance. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817, Brussels, Belgium. Association for Computational Linguistics.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019a. Analyzing sentence fusion in abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 104–110, Hong Kong, China. Association for Computational Linguistics.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Lidan Wang, Walter Chang, and Fei Liu. 2020. Understanding points of correspondence between sentences for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Seattle, United States. Association for Computational Linguistics.

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019b. Scoring sentence singletons and pairs for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2175–2189, Florence, Italy. Association for Computational Linguistics.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196, Florence, Italy. 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. Association for Computational Linguistics.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. 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*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Yashar Mehdad, Giuseppe Carenini, Frank Tompa, and Raymond T. Ng. 2013. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 136–146, Sofia, Bulgaria. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv:1910.10683*.

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. Association for Computational Linguistics.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada. Association for Computational Linguistics.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond SumBasic: Task-focused Summarization with Sentence Simplification and Lexical Expansion. *Information Processing and Management*, 43(6):1606–1618.

Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D. Manning, and Curtis P. Langlotz. 2020. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of the 58th Annual Conference of the Association for Computational Linguistics (ACL)*.