Extractive Entity-Centric Summarization as Sentence Selection using Bi-Encoders

Ella Hofmann-Coyle[*1], **Mayank Kulkarni**[*†2], **Lingjue Xie**[*1]
Mounica Maddela[†3], **Daniel Preoţiuc-Pietro**[1]
[1]Bloomberg [2]Amazon Alexa AI [3]Georgia Institute of Technology
ehofmanncoyl@bloomberg.net, maykul@amazon.com, lxie91@bloomberg.net
mmadela3@cc.gatech.edu, dpreotiucpie@bloomberg.net

Abstract

Entity-centric summarization is a type of controllable summarization that aims to produce a summary of a document that is specific to a given target entity. Extractive summaries possess multiple advantages over abstractive ones such as preserving factuality and can be directly used in downstream tasks like target-based sentiment analysis or incorporated into search applications. In this paper, we explore methods to solve this task by recasting it as a sentence selection task, as supported by the EntSUM data set. We use methods inspired by information retrieval, where the input to the model is a pair representing a sentence from the original document and the target entity, in place of the query. We explore different architecture variants and loss functions in this framework with results showing an up to 5.8 F1 improvement over past state-of-the-art and outperforming the competitive entity-centric Lead 3 heuristic by 1.1 F1. In addition, we also demonstrate similarly strong results on the related task of salient sentence selection for an entity.

1 Introduction

Controllable summarization is a recently growing area of research, where the aim is to provide a summary that is specific to a user's information need, which could be a target entity (Maddela et al., 2022), aspect (Amplayo et al., 2021) or topic – or can represent the user's preferred style (Fan et al., 2018) or length (Kikuchi et al., 2016; Dou et al., 2021). Controllable summarization offers the promise of making summarization more usable to users, enabling them to achieve their end goals by summarizing the information they are interested in (Jones, 1999).

Extractive summarization aims to extract passages or entire sentences from the original summaries, as opposed to abstractive summarization which aims to generate an entirely new summary (Nenkova et al., 2011). Although most recent research has focused on abstractive summarization techniques, these possess several disadvantages, the most prominent being the potential for lack of factuality and coherence (Cao et al., 2018; Kryscinski et al., 2019; Lebanoff et al., 2019), as well as difficulty in correctly assessing the summary quality automatically (Rankel et al., 2013; Peyrard, 2019; Zhang et al., 2019). On the other hand, extractive summarization mitigates these issues by extracting text from the original document and, if the data set contains the sentence or passage level information, evaluation can then be performed using standard metrics, such as F1. The extractive entity-centric summarization methods can be used directly to aid users in interactive applications such as search (Varadarajan and Hristidis, 2006; Turpin et al., 2007), through either highlighting or extracting passages in the document. Extractive summarization also has the potential to be used as an intermediary step or auxiliary task in downstream entity-centric tasks, such as entity salience (Gamon et al., 2013; Dunietz and Gillick, 2014), aspect-based sentiment classification (Pontiki et al., 2016), or information retrieval.

This paper presents the first in-depth study of extractive entity-centric summarization methods. We take advantage of the unique properties of the EntSUM data set (Maddela et al., 2022), which provides multiple layers of annotations regarding the entities, including the sentences salient for the entity in a document and the sentences that construct a summary about an entity. We are thus able to recast the entity-centric extractive summarization task as selecting the summary sentences regarding an entity in a document. This allows us to compute reliable F1 metrics to compare several approaches, including heuristics and adaptations to extractive summarization of controllable summarization methods. We propose new meth-

[*] The authors contributed equally and are listed in alphabetical order. [†] The work was done while at Bloomberg.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 326–333
November 20–23, 2022.

ods for entity-centric summarization using the bi-encoder framework with pre-trained Transformer-based models which significantly outperform past approaches to entity-centric summarization and further, outperform the challenging entity-centric lead-3 baseline in summarization tasks.

Our contributions are (1) framing the entity-centric summarization task as sentence selection; (2) a new state-of-the-art method for the task; (3) data analysis for insight into model behavior.

2 Data

We use the EntSUM data set introduced in (Maddela et al., 2022) to evaluate our methods. The EntSUM data set consists of 2,788 entity-centric summaries across 645 documents annotated on top of the test split of the New York Times (NYT) (Sandhaus, 2008) summarization data set. In this paper, we use 2 out of the 4 annotations in EntSUM: the salient sentences to the entity and the summary sentences for an entity.

Each entity is mentioned on average in 3.95 sentences. Annotators labeled as salient sentences to the entity all sentences relevant to a given entity with an average of 5.8 sentences/entity. Annotators selected sentences to compose the entity-centric summary as a subset of the salient sentences, resulting in an average of 2.49 sentences.

3 Task Definition

We define the task of extractive entity-centric summarization as selecting a set of sentences $\{S_1^e \dots S_k^e\}$ from a single document $D = \{S_1 \dots S_n\}$, when given the document D and a target entity e as input. This type of problem formulation is facilitated by the EntSUM data set, as explicit annotations exist at the sentence level. This also allows us to use standard precision, recall and F1 metrics for the evaluation of extractive summarization.

To date, sentence-level classification was rare due to the complexity and resource-intensive nature of obtaining the annotations. Most large single-document summarization data sets have been collected by aligning full documents with a hand-written abstractive summary obtained from titles (Narayan et al., 2018), bullet points (Hermann et al., 2015) summaries created for indexing purposes (Sandhaus, 2008) or TL;DR's created by scientific paper authors (Cachola et al., 2020). The lack of sentence-level annotations required previous extractive summarization methods (See et al., 2017; Liu and Lapata, 2019; Zhong et al., 2020) to be trained on greedily generated weak sentence-level labels obtained using content overlap metrics such as ROUGE (Lin, 2004) or were evaluated on abstractive summaries using overlap measures such as ROUGE or BERTScore (Zhang et al., 2019), which at times are unable to properly capture semantic similarity. This type of evaluation and setup is more common in multi-document extractive summarization research (Kim et al., 2011; Angelidis and Lapata, 2018; Amplayo and Lapata, 2021; Angelidis et al., 2021). Evaluation using F1 is arguably more reliable and less ambiguous, albeit there are also some caveats associated with using this task setup such as granularity (Nenkova et al., 2011).

4 Methods

We experiment with the categories of methods listed below. Methods with Ent in their name identify sentences containing the target entity and restrict inference to only those sentences. Entities are identified by using the Flair NER model (Akbik et al., 2018) and SpanBERT (Joshi et al., 2020) for coreference resolution, then matched to the target entity using string matching.

4.1 Heuristics

LeadK-Overall is a generic summarization method that selects the first k sentences in the document regardless of the target entity.
LeadK-Ent uses the entity detection pipeline to identify the first k sentences in a document with a given entity. This is a strong heuristic corresponding to the LeadK method for generic summaries, which relies on the fact that the first few sentences contain salient information (Nallapati et al., 2017).
All-Ent uses the entity detection pipeline to identify *all* sentences in a document with a given entity.

4.2 Oracle Methods

Oracle methods use annotations for a given task to provide an upper bound to a series of methods.
LeadK-Oracle-Salient selects the top k sentences from the gold salient sentence annotations.
LeadK-Oracle-Summary selects the top k sentences from the gold summary sentence annotations.

4.3 BERTSum Variants

In line with the original EntSUM paper Maddela et al. (2022), we use extractive methods based on

the state-of-the-art extractive generic summarization architectures of BERTSum (Liu and Lapata, 2019). Sentence representations are generated for each sentence through a BERT encoder (Devlin et al., 2019) Interactions between these sentences are modeled through a summarization layer, which generates a representation for each sentence that is passed to a classifier to determine if the sentence should be added to the summary. We choose up to three sentences to control for summary length when compared with the Lead3 methods.

BERTSum-Overall is the BERTSum model for generic summarization.

BERTSum-Ent is an adaptation of BERTSum which only uses the entity detection pipeline as input and is trained on proxy summaries. This is the best performing extractive method from Maddela et al. (2022).

BERTSum-Prefix adds the target entity as a prefix to the input document, which is then passed to BERTSum-Ovr. This is inspired by entity prepending in controllable abstractive summarization (Fan et al., 2018; He et al., 2020) and extractive aspect-oriented opinion summarization (Ahuja et al., 2022).

BERTSum-Coref-Prefix replaces the BERT encoder weights in BERTSum-Prefix with pre-trained SpanBERT-coref[1] encoder weights with the aim on enhancing the input with coreference information.

4.4 Bi-Encoders

The bi-encoder architecture takes an input pair and uses two encoders to represent the two inputs independently as dense vectors. Training is done by taking a loss function involving the two vectors and the gold label, such as a cosine similarity loss. At inference time, a similarity metric is computed across the two representations. Bi-encoders using Transformer-based pre-trained language models have achieved state-of-the-art results in many tasks that operate on pairs such as entity linking (Wu et al., 2020), sentence similarity (Reimers and Gurevych, 2019) or passage retrieval (Karpukhin et al., 2020).

We experiment with the following versions using BERT as the encoder in all cases:

Encoder types: we experiment with both having the same encoder updated by both inputs (**Tied**) and updated independently (**Untied**) when training on the pair classification task.

Loss Functions: we use cosine similarity (**Cos**) or contrastive loss (**Cntr**). Cosine similarity is computed between the entity (e) and sentence (s) representations and the binary label Y is used in the loss defined as $L_{cos} = \|Y - \frac{e.s}{\|e\|\|s\|}\|_2$. Contrastive loss (Hadsell et al., 2006) requires similar pairs S and dissimilar pairs D to define the loss function as $L_{con} = (1-Y)L_S + YL_D$, for a given label $Y \in \{0, 1\}$ with the goal of maximizing the margin between the positive and negative sample boundary.

Sentence selection: we select sentences for the summary either by thresholding on the cosine similarity value (here, 0.5) between the target entity and all sentences in the document (**Thres**) or by taking the top k values (**Top**) above the threshold.

We experimented with adapting the BERTSum architecture to a bi-encoder setup, however, the results are underwhelming and are omitted for brevity.

5 Experimental Setup

5.1 Training

We follow the experimental setup of Maddela et al. (2022), where we train on the NYT data set without entity-centric annotations and use the annotated EntSUM data set only for testing. We thus create training data by creating weak labels for entity-centric summaries from generic ones.

We train all our methods on the NYT corpus consisting of 44,382 training and 5,523 validation (document, summary) pairs as specified in (Kedzie et al., 2018). This data set size increases to 464,339 training and 58,991 validation pairs when training in the BERTSum setup as each document contains multiple entities resulting in multiple document summary pairs for a single document. This is further extended to 16,710,624 training and 2,152,164 validation samples in the bi-encoder setup as the training is done at a sentence level.

We use the first three sentences in the source texts containing the entity as the gold training summary. We only add the sentence to the gold summary if the fuzz ratio in fuzzy string matching[2] is less than 60 with the existing sentences in the summary to avoid duplication in meaning. For the bi-encoder experiments, these sentences in the summary are paired with the entity to be considered as positive

[1] https://github.com/mandarjoshi90/coref

[2] https://github.com/seatgeek/thefuzz

examples, while all sentences not part of the summary are treated as negative examples.

For heuristic methods and selecting top sentences, we use $k = 3$ for the summarization task and $k = 6$ for salient sentence selection. These values were set using the summary statistics of the data set.

5.2 Hyperparameters

We follow the hyperparameters and implementation described in the BERTSum[3] for all the BERTSum variants. In the bi-encoder experiments, we train the model for 2 epochs with batch size 8 and use 10% of train data for warm-up. We use default hyperparameter values specified in the sentence-transformers repository.[4]

5.3 Evaluation

We evaluate our methods using the F1 score, as the prediction is at the sentence level. The EntSUM data set contains 867 examples that contain two annotations for the same entity, which were collected for quality assurance purposes. For thresholding, use $k = 3$ for summarization and no constraints for salient sentence selection. We use the following method to compute the F1 score against both references as follows:

- we evaluate the model independently on each of the annotations;
- we average the F1 score across the two annotations and assign this score to this example;
- these scores are then combined with the scores obtained for the rest of the 1,921 single annotations to obtain a score on the entire data set.

6 Results

Table 1 shows the performance of all the proposed methods on extractive summarization (**Summary**), as well as the upstream task of salient sentence extraction, which aims to identify all sentences relevant to a target entity. Our findings are:

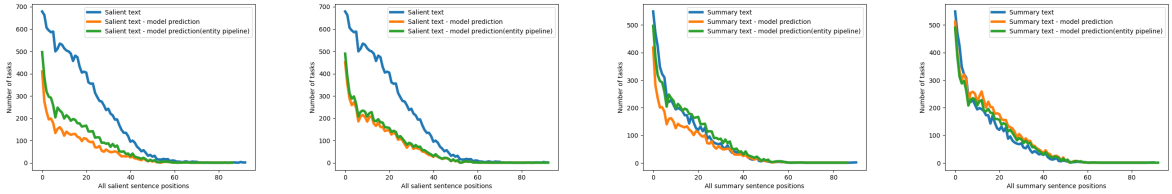
- Bi-encoder (BE) methods obtain the best results, with 5.9 F1 above the past state-of-the-art method (BERTSum-Ent) and, moreover, outperforms the strong Lead3 heuristic by 1.1 F1 (Lead3-Ent).
- Inference using the entity identification pipeline is necessary for high performance, with the best method not using this being 11.2 F1 lower than the best results.

[3] https://github.com/nlpyang/BertSum
[4] https://www.sbert.net/

Model	Salient	Summary
Lead3-Overall	15.2	16.9
Lead3-Ent	51.5	72.0
Lead6-Ent	63.6	67.4
All-Ent	**77.9**	62.9
BERTSum-Overall	15.3	17.4
BERTSum-Ent	–	67.2
BERTSum-Prefix	18.9	19.3
BERTSum-Coref-Prefix	31.2	24.2
BE-Cos-Tied-Thres	52.5	60.4
BE-Cos-Tied-Top	55.6	57.3
BE-Cos-Tied-Ent-Thres	–	61.1
BE-Cos-Tied-Ent-Top	–	<u>73.0</u>
BE-Cos-Untied-Thres	49.4	54.9
BE-Cos-Untied-Top	54.9	56.4
BE-Cos-Untied-Ent-Thres	–	55.6
BE-Cos-Untied-Ent-Top	–	72.7
BE-Cntr-Tied-Thres	<u>70.7</u>	61.9
BE-Cntr-Tied-Top	55.9	57.6
BE-Cntr-Tied-Ent-Thres	–	71.9
BE-Cntr-Tied-Ent-Top	–	**73.1**
Lead3-Oracle-Salient	56.1	74.4
Lead6-Oracle-Salient	79.8	76.6
Lead3-Oracle-Summary	52.5	85.8

Table 1: Results in F1 score on the EntSUM data set for the tasks of salient sentence selection (**Salient**) and extractive summarization (**Summary**). **Bold** and <u>underline</u> indicate the best and second best performing models. Oracle methods use gold annotations and are excluded from the best results.

- Results using oracle methods show that, given gold salient sentences, the performance is close to the best method (+1.3 F1), while the Lead3 method with gold summary sentences is 12.7 F1 higher. This shows that the remaining performance gain is to be had by a better ranking of salient sentences, even when constrained to always selecting the top 3 sentences, rather than the ability to retrieve these from the non-salient ones. Note the gap between Lead3-Oracle-Summary performance and 100 F1 is caused by summaries that contain fewer than 3 sentences.
- Bi-encoders with untied encoders are less effective than sharing weights even in this asymmetric setting. We believe the reason for this is that the entity names as queries are fairly short and the skewed ratio of 1:22 pairs of positive and negative sentences makes it difficult for an independent encoder to learn a rich representation of the entity space.



(a) Salient sentence prediction Cosine Loss (b) Salient sentence prediction Contrastive Loss (c) Summary sent. prediction Cosine Loss (d) Summary sent. prediction Contrastive Loss

Figure 1: Distribution of sentence position predictions.

- The loss function choice does not have a very large impact on the results, with the contrastive loss achieving slightly better results.
- Methods that prepend the entity to the document slightly outperform the entity agnostic methods, but are over 40 F1 lower than bi-encoder approaches, demonstrating the inefficiency of this type of approach.
- Entity agnostic summaries show performance under 20 F1, highlighting the large gap between the generic and controlled summarization tasks.

6.1 Salient Sentence Selection

We test our methods for summarization on the salient sentence selection task to probe the extent to which our methods are able to capture the entity - sentence association, in addition to understanding the importance of the sentence to the summary. Table 1 shows that, despite not being trained for this task, the best performing method performs better than many heuristic-based methods (Lead3-Ent, Lead6-Ent, Lead3-Oracle-Salient) and is only 9.1 F1 lower than taking the top 6 sentences annotated as being salient to the entity, where 6 is the closest integer value to the average number of salient sentences. Training with contrastive loss is more effective at capturing entity-sentence relationship (e.g. +18.2 F1 for BE-Cos-Tied-Thres vs BE-Cntr-Tied-Thres) even if overall summarization performance is similar (+1.5 F1). Note that the methods using the entity detection pipeline are not evaluated on the salient sentence selection task.

6.2 Model Prediction Analysis

Finally, we analyze the positions within the document of the predictions compared to gold labels for both summary and salient sentence selection tasks across the two bi-encoder loss functions and with or without using the entity extraction pipeline.

Figures 1a and 1b compare the salient sentence task predictions with the two losses. We plot the distribution of sentence position predictions to identify patterns where the models over/under predict. We see that the number of sentences predicted in the first half of the document is fewer, we conjecture this is because fewer sentences exhibit a high similarity score and because we also truncate to the top 3 sentences if more are predicted. We see in Figure 1a that the model prediction with Cosine Similarity Loss is slightly underperforming the *ent* pipeline, however, these differences are largely reconciled when using the Contrastive Loss in Figure 1b where the lines almost overlap.

Figures 1c and 1d compare the models for summary sentence prediction when using the cosine similarity and contrastive losses. We note that the *ent* pipeline performs fairly well in being able to predict the summary sentences with a high overlap with the actual summary sentences. We observe an interesting phenomenon when using Cosine Similarity Loss as seen in Figure 1c where the model predicts fewer summary sentences at the beginning of the document but aligns well with the summary sentences close to the middle of the document. However, when using contrastive loss, more summary sentences are predicted at the beginning of the document and also across the rest of the document, resulting in higher recall and thus improving downstream performance.

7 Conclusions

This paper explored the task of entity-centric extractive summarization. Results showed that by leveraging sentence encoders in a bi-encoder architecture, we are able to substantially outperform previous controllable extractive summarization methods and the competitive Lead3 heuristic. This method also performs well without adaptations in the auxiliary task of salient sentence extraction. Future work can investigate how best to build entity representations, custom loss functions for this task and joint sentence selection across the entire document.

References

Ojas Ahuja, Jiacheng Xu, Akshay Gupta, Kevin Horecka, and Greg Durrett. 2022. ASPECTNEWS: Aspect-oriented summarization of news documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6494–6506, Dublin, Ireland. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Aspect-controllable opinion summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6578–6593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reinald Kim Amplayo and Mirella Lapata. 2021. Informative and controllable opinion summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2662–2672, Online. Association for Computational Linguistics.

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Thirty-second AAAI Conference on Artificial Intelligence*, AAAI.

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.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Jesse Dunietz and Daniel Gillick. 2014. A new entity salience task with millions of training examples. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 205–209, Gothenburg, Sweden. Association for Computational Linguistics.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Michael Gamon, Tae Yano, Xinying Song, Johnson Apacible, and Patrick Pantel. 2013. Identifying salient entities in web pages. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2375–2380.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Fatema Rajani, and Caiming Xiong. 2020. Ctrlsum: Towards generic controllable text summarization. *CoRR*, abs/2012.04281.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701.

Karen Sparck Jones. 1999. Automatic summarizing: factors immarizing: factors and directions. *Advances in automatic text summarization*, page 1.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

Hyun Duk Kim, Kavita Ganesan, Parikshit Sondhi, and ChengXiang Zhai. 2011. Comprehensive review of opinion summarization.

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.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. 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.

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.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Mounica Maddela, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2022. EntSUM: A data set for entity-centric extractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3355–3366, Dublin, Ireland. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI Conference on Artificial Intelligence*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.

Maxime Peyrard. 2019. Studying summarization evaluation metrics in the appropriate scoring range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100, Florence, Italy. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Peter A. Rankel, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2013. A decade of automatic content evaluation of news summaries: Reassessing the state of the art. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 131–136, Sofia, Bulgaria. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

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.

Andrew Turpin, Yohannes Tsegay, David Hawking, and Hugh E Williams. 2007. Fast generation of result snippets in web search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 127–134.

Ramakrishna Varadarajan and Vagelis Hristidis. 2006. A system for query-specific document summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 622–631.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*, ICLR.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.