Unsupervised Abstractive Dialogue Summarization with Word Graphs and POV Conversion

Seongmin Park, Jihwa Lee
ActionPower, Seoul, Republic of Korea
{seongmin.park, jihwa.lee}@actionopwer.kr

Abstract

We advance the state-of-the-art in unsupervised abstractive dialogue summarization by utilizing multi-sentence compression graphs. Starting from well-founded assumptions about word graphs, we present simple but reliable path-reranking and topic segmentation schemes. Robustness of our method is demonstrated on datasets across multiple domains, including meetings, interviews, movie scripts, and day-to-day conversations. We also identify possible avenues to augment our heuristic-based system with deep learning. We open-source our code[1], to provide a strong, reproducible baseline for future research into unsupervised dialogue summarization.

1 Introduction

Compared to traditional text summarization, dialogue summarization introduces a unique challenge: conversion of first- and second-person speech into third-person reported speech. Such discrepancy between the observed text and expected model output puts greater emphasis on abstractive transduction than in traditional summarization tasks. The disorientation is further exacerbated by each of many diverse dialogue types calling for a differing form of transduction – short dialogues require terse abstractions, while meeting transcripts require summaries by agenda.

Thus, despite the steady emergence of dialogue summarization datasets, the field of dialogue summarization is still bottlenecked by a scarcity of training data. To train a truly robust dialogue summarization model, one requires transcript-summary pairs not only across diverse *dialogue domains*, but also across multiple *dialogue types* as well. The lack of diverse annotated summarization data is especially pronounced in low-resourced languages. From such state of the literature, we identify a need for unsupervised dialogue summarization.

[1]https://github.com/seongminp/graph-dialogue-summary

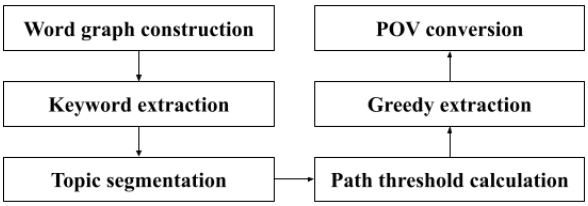

Figure 1: Our summarization pipeline.

Our method builds upon previous research on unsupervised summarization using word graphs. Starting from the simple assumption that *a good summary sentence is at least as informative as any single input sentence*, we develop novel schemes for path extraction from word graphs. Our contributions are as follows:

1. We present a novel scheme for path reranking in graph-based summarization. We show that, in practice, simple keyword counting performs better than complex baselines. For longer texts, we present an optional topic segmentation scheme.

2. We introduce a point-of-view (POV) conversion module to convert semi-extractive summaries into fully abstractive summaries. The new module by itself improves all scores on baseline methods, as well as our own.

3. Finally, We verify our model on datasets beyond those traditionally used in literature to provide a strong baseline for future research.

With just an off-the-shelf part-of-speech (POS) tagger and a list of stopwords, our model can be applied across different types of dialogue summarization.

2 Background

2.1 Multi-sentence compression graphs

Pioneered by Filippova (2010), a Multi-Sentence Compression Graph (MSCG) is a graph whose

Proceedings of the 2nd Workshop on Deriving Insights from User-Generated Text, pages 1 - 9

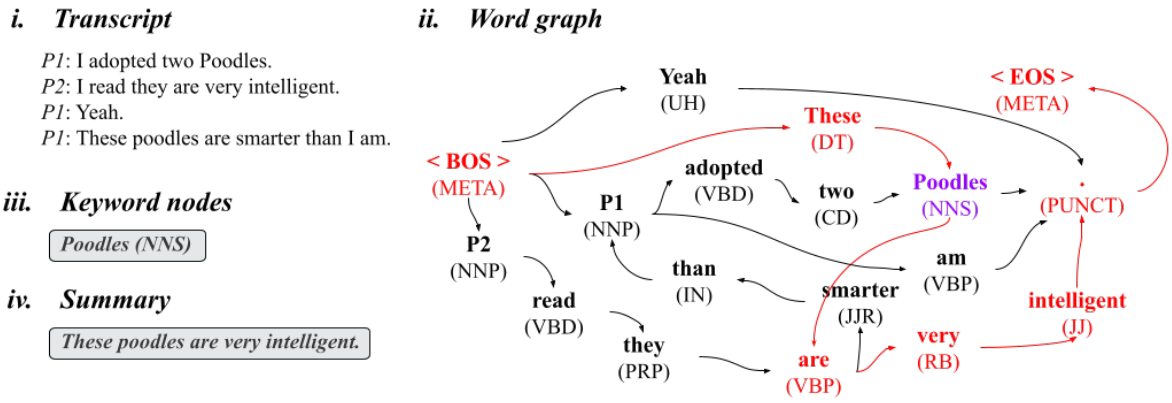

Figure 2: Construction of word graph. Red nodes and edges denote the selected summary path. Node highlighted in purple ("Poodles") is the only non-stopword node included in the k-core subgraph of the word graph. We use nodes from the k-core subgraph as keyword nodes. All original sentences from the unabridged input is present as a possible path from v_{bos} to v_{eos}. Paths that contain more information than those original paths are extracted as summaries.

nodes are words from the input text and edges are coocurrance statistics between adjacent words. During preprocessing, words "<bos>" (beginning-of-sentence) and "<eos>" (end-of-sentence) are prepended and appended, respectively, to every input sentence. Thus, all sentences from the input are represented in the graph as a single path from the *<bos>* node (v_{bos}) to the *<eos>* node (v_{eos}). Overlapping words among sentences will create intersecting paths within MSCG, creating new paths from v_{bos} to v_{eos}, unseen in the original text. Capturing these possibly shorter but informative paths is the key to performant summarization with MSCGs.

Ganesan et al. (2010) introduce an abstractive sentence generation method from word graphs to produce opinion summaries. Tixier et al. (2016) show that nodes with maximal neighbors – a concept captured by graph degeneracy – likely belong to important keywords of the document. Shortest paths from v_{bos} to v_{eos} are scored according to how many keyword nodes they contain. Subsequently, a budget-maximization scheme is introduced to find the set of paths that maximizes the score sum within designated word count (Tixier et al., 2017). We also adopt graph degeneracy to identify keyword nodes in MSCG.

2.2 Unsupervised Abstractive Dialogue Summarization

Aside from MSCGs, unsupervised dialogue summarization usually employ end-to-end neural architectures. Zhang et al. (2021) and Zou et al. (2021) utilize text variational autoencoders (VAEs) (Kingma and Welling, 2014; Bowman et al., 2016) to decode conditional or denoised abridgements. Fu et al. (2021) reformulate summary generation into a self-supervised task by equipping auxiliary objectives to the training architecture. Among end-to-end frameworks we only include Fu et al. (2021) as our baseline, because the brittle nature of training text VAEs, coupled with the lack of detail on data and parameters used to train the models, render VAE-based methods beyond reproducible.

3 Summarization strategy

In following subsections we outline our proposed summarization process.

3.1 Word graph construction

First, we assemble a word graph G from the input text. We use a modified version of Filippova (2010)'s algorithm for graph construction:

- Let SW be a set of stopwords and $T = s_0, s_1, ...$ be a sequence of sentences in the input text.
- Decompose all $s_i \in T$ into a sequence of POS-tagged words.

$$s_i = (\text{“}bos\text{”}, \text{“}meta\text{”}), (w_{i,0}, pos_{i,0}), ..., (w_{i,n-1}, pos_{i,n-1}), (\text{“}eos\text{”}, \text{“}meta\text{”}) \quad (1)$$

- For every $(w_{i,j}, pos_{i,j}) \in s_i$ such that $w_i \notin SW$ and $s_i \in T$, add a node v in G. If a

node v' with the same lowercase word $w_{i,k}$ and tag $pos_{i,k}$ such that $j \neq k$ exists, pair $(w_{i,j}, pos_{i,j})$ with v' instead of creating a new node. If multiple such matches exist, select the node with maximal overlapping context ($w_{i,j-1}$ and $w_{i,j+1}$).

- Add stopword nodes – $(w_{i,j}, pos_{i,j}) \in s_i$ such that $w_{i,j} \in SW$ and $s_i \in T$ – to G with the algorithm described above.

- For all $s_i \in T$, add a directed edge between node pairs that correspond to subsequent words. Edge weight w between nodes v_1 and v_2 is calculated as follows:

$$w' = \frac{freq(v_1) + freq(v_2)}{(\sum_{s_i \in T} diff(i, v_1, v_2))^{-1}} \quad (2)$$

$$w'' = freq(v_1) * freq(v_2) \quad (3)$$

$$w = w'/w'' \quad (4)$$

$freq(v)$ is the number of words from original text mapped to node v. $diff(i, v_1, v_2)$ is the absolute difference in word positions of v_1 and v_2 within s_i:

$$diff(i, v_1, v_2) = |k - j| \quad (5)$$

, where w_{ij} and w_{ik} are words in s_i that correspond to nodes v_1 and v_2, respectively.

In edge weight calculation, w' favors edges with strong cooccurrence, while w''^{-1} favors edges with greater salience, as measured by word frequency.

It follows from above that only a single <*bos*> node and a single <*eos*> node will exist once the graph is completed.

3.2 Keyword extraction

The resulting graph from the previous step is a composition that captures syntactic importance. Traditional approaches utilize centrality measures to identify important nodes within word graphs (Mihalcea and Tarau, 2004; Erkan and Radev, 2004). In this work we use graph degeneracy to extract keyword nodes. In a k-degenerate word graph, words that belong to k-core nodes of the graph are considered to be keywords. We collect KW, a set of nodes belonging to the k-core subgraph. The k-core of a graph is the maximally degenerate subgraph, with minimum degree of at least k.

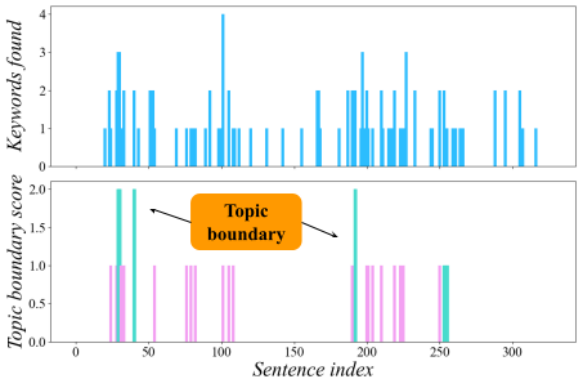

Figure 3: Topic segmentation on AMI meeting ID ES2005b. Green bars indicate sentence boundaries with highest topic distance.

3.3 Path threshold calculation

Once keyword nodes are identified, we score every path from v_{bos} to v_{eos} that corresponds to a sentence from the original text. Contrary to previous research into word-graph based summarization, we use a simple keyword coverage score for every path:

$$Score_i = \frac{|V_i \cap KW|}{|KW|} \quad (6)$$

, where V_i is the set of all nodes in path p_i, a representation of sentence $s_i \in T$, within the word graph. We calculate the path threshold t, the mean score of all sentences in the original text. Later, when summaries are extracted from the word graph, candidates with path score less than t are discarded. We also experimented with setting t as the minimum or maximum of all original path scores, but such configurations yielded inferior summaries influenced by outlier path scores.

Our path score function is reminiscent of the diversity reward function in Shang et al. (2018). However, we use the function as a measure of *coverage* instead of *diversity*. More importantly, we utilize the score as means to extract a threshold based on all input sentences, which is significantly different from Shang et al. (2018)'s utilization of the function as a monotonically increasing scorer in submodularity maximization.

3.4 Topic segmentation

For long texts, we apply an optional topic segmentation step. Our summarization algorithm is separately applied to each segmented text. Similar to path ranking in the next section, topics are determined according to keyword frequency. For every

Dataset	Domain	Test files	Dialogue length (chars)	Summary length (chars)
AMI	Meeting	20	22,499 (4,665 words)	1,808 (292 words)
ICSI	Meeting	6	42,484 (8,926 words)	2,271 (371 words)
DialogSum	Day-to-day	500	633 (125 words)	115 (19 words)
SAMSum	Day-to-day	819	414 (84 words)	109 (20 words)
MediaSum	Interview	10,000	8,718 (1,562 chars)	335 (59 words)
SummScreen	Screenplay	2,130	23,693 (5,642 words)	1,795 (342 words)
ADS	Debate	45	2918 (534 words)	882 (150 words)

Table 1: Statistics for benchmark datasets. All character-level and word-level statistics are averaged over the test set and rounded to the nearest whole number.

sentence in the input, we construct a *topic coverage* vector c, a zero-initialized row-vector of length $|KW|$. Each column of the row vector is a binary representation signaling the presence of a single element in KW. Topic coverage vector of a path containing two keywords from KW, for instance, would contain two columns with 1.

Every transition between sentences is a potential topic boundary. Since each sentence (and corresponding path) has an associated topic coverage vector, we quantify the topic distance d of a sentence with the next as the negative cosine distance of their topic vectors:

$$d_{i,i+1} = -\frac{c_i \cdot c_{i+1}}{\|c_i\|\|c_{i+1}\|} \tag{7}$$

If p is a hyperparameter representing the total number of topics, one can segment the original text at $p-1$ sentence boundaries with the greatest topic distance. Alternatively, sentence boundaries with topic distance greater than a designated threshold can be selected as topic boundaries. For simplicity, we proceed with the former segmentation setup (top-p boundary) when necessary.

3.5 Summary path extraction

We generate a summary per-speaker. Our construction of the word graph allows fast extraction of sub-graphs containing only nodes pertaining to utterances from a single speaker. For each speaker subgraph, we generate summary sentences as follows:

1. We obtain k shortest paths from v_{bos} to v_{eos} by applying the k-shortest paths algorithm (Yen, 1971) to our word graph.

2. Iterating from the shortest path, we collect any paths with keyword coverage score above the threshold calculated in 3.3.

3. For each path found, we track the set of encountered keywords in KW. We stop our search if all keywords in KW were encountered, or a pre-defined number of iterations (the search depth) is reached.

A good summary has to be both concise and informative. Intuitively, edge weights of the proposed word graph captures the former, while keyword thresholding prioritizes the latter.

3.6 POV conversion

Finally, we convert our collected semi-extractive summaries into abstractive reported speech using a rule-based POV conversion module. We describe sentences extracted from our word graph as *semi-extractive* rather than *extractive*, to recognize the distinction between previously unseen sentences created from pieces of text, and sentences taken verbatim from the original text. Similar to existing *extract-then-abstract* summarization pipelines (Mao et al., 2021; Liu et al., 2021), our method hinges on the assumption that the extractive path-reranking step will optimize for *summary content*, while the succeeding abstractive POV-conversion step will do so for *summary style*. FewSum (Bražinskas et al., 2020) also applies POV conversion in a few-shot summarization setting. FewSum conditions the summary generator to produce sentences in targeted styles, which is achieved by nudging the decoder to generate pronouns appropriate for each designated tone.

Popular literature has established that defining an all-encompassing set of rules for indirect speech conversion is infeasible (Partee, 1973; Li, 2011). In fact, the English grammar is mostly descriptive rather than prescriptive – no set of official rules dictated by a single governing authority exists. Even so, rule based POV conversion does provide a strong baseline compared to state-of-the-art

Model	AMI			ICSI		
	R1	R2	RL	R1	R2	RL
RepSum Fu et al. (2021)	18.88	2.38	15.62	-	-	-
Filippova (2010)	33.47	6.21	15.15	26.53	3.69	12.09
Mehdad et al. (2013)	34.62	6.49	15.41	27.20	3.57	12.55
Boudin and Morin (2013)	34.21	6.37	14.92	26.90	3.64	12.18
Shang et al. (2018)	34.34	6.13	15.58	26.93	3.65	12.68
Filippova (2010) $_{+POV}$	34.16	6.35	15.27	26.79	3.81	12.21
Mehdad et al. (2013) $_{+POV}$	**35.39**	6.59	15.54	27.48	3.65	12.66
Boudin and Morin (2013) $_{+POV}$	34.93	6.49	15.07	27.14	3.72	12.20
Shang et al. (2018) $_{+POV}$	34.91	6.18	**15.70**	27.27	3.72	**12.78**
Ours $_{PreSeg}$	32.21	5.55	14.85	27.60	4.43	11.66
Ours $_{TopicSeg}$	33.30	6.59	14.19	27.66	4.27	12.16
Ours $_{PreSeg+POV}$	33.66	**6.85**	14.17	27.80	**4.56**	11.77
Ours $_{TopicSeg+POV}$	33.21	5.84	15.30	**27.84**	4.33	12.29

Table 2: Results on meeting summarization datasets. All reported scores are F-1 measures. Models with POV indicate post-proceessing with our suggested POV conversion module. $PreSeg$ models utilize topic segmentations provided in Shang et al. (2018), and $TopicSeg$ models intake unsegmented raw transcripts and perform the topic segmentation algorithm suggested in this paper. Results for RepSum are quoted from the original paper.

techniques, such as end-to-end Transformer networks (Lee et al., 2020). In this study, we limit our scope to rule-based conversion because only the rule-based system among all tested methods in Lee et al. (2020) confers to the unsupervised nature of this paper. We encourage further research into integrating more advanced reported speech conversion techniques into the abstractive summarization pipeline.

In this work, we apply four conversion rules:

1. Change pronouns from first person to third person.
2. Change modal verbs *can*, *may*, and *must* to *could*, *might*, and *had to*, respectively.
3. Convert questions into a pre-defined template: <*Speaker*> *asks* <*utterance*>.
4. Fix subject-verb agreement after applying rules above.

We notably omit prepend rules suggested in (Lee et al., 2020), because the input domain of our summarization system is unbounded, unlike with task-oriented spoken commands for virtual assistants. We also leave tense conversion for future research.

4 Experiments

4.1 Datasets

We test our model on dialogue summarization datasets across multiple domains:

1. Meetings: *AMI* (McCowan et al., 2005), *ICSI* (Janin et al., 2003)
2. Day-to-day conversations: *DialogSum* (Chen et al., 2021b), *SAMSum* (Gliwa et al., 2019)
3. Interview: *MediaSum* (Zhu et al., 2021)
4. Screenplay: *SummScreen* (Chen et al., 2021a)
5. Debate: *ADS* (Fabbri et al., 2021)

Table 1 provides detailed statistics and descriptions for each dataset.

For AMI and ICSI, we conduct several ablation experiments with different components of our model omitted: semi-extractive summarization without POV conversion is compared with fully-abstractive summarization with POV conversion; utilization of pre-segmented text provided by Shang et al. (2018) is compared with application of topic segmentation suggested in this paper.

4.2 Baselines

For meeting summaries, we compare our method with previous research on unsupervised dialogue summarization. Along with Filippova (2010), Shang et al. (2018), and Fu et al. (2021), we select Boudin and Morin (2013) and Mehdad et al. (2013) as our baselines. All but Fu et al. (2021) are word graph-based summarizers.

For all other categories, we choose LEAD-3 as our unsupervised baseline. LEAD-3 selects the

Dataset	Our results			LEAD-3		
	R1	R2	RL	R1	R2	RL
DialogSum	**20.79**	5.43	15.14	19.46	**6.19**	**15.99**
SAMSum	**26.48**	**9.69**	**19.65**	21.93	8.52	18.65
MediaSum	7.19	1.79	5.66	**8.58**	**3.19**	**6.62**
SummScreen	**21.25**	**2.23**	**9.40**	5.18	0.55	3.75
ADS	**28.00**	**7.33**	**14.75**	19.39	5.72	13.22

Table 3: Results on day-to-day, interview, screenplay, and debate summarization datasets. All reported scores are F-1 measures. In our method, topic segmentation is applied to datasets with average transcription length greater than 5,000 characters (MediaSum, SummScreen), and POV conversion is applied to all datasets.

first three sentences of a document as the summary. Because summary distributions in several document types tend to be front-heavy (Grenander et al., 2019; Zhu et al., 2021), LEAD-3 provides a competitive extractive baseline with negligible computational burden.

4.3 Evaluation

We evaluate the quality of generated system summaries against reference summaries using standard ROUGE scores (Lin, 2004). Specifically, we use ROUGE-1 ($R1$), ROUGE-2 ($R2$), and ROUGE-L (RL) scores that respectively measure unigram, bigram, and longest common subsequence coverage.

5 Results

5.1 Meeting summarization

Table 2 records experimental results on AMI and ISCI datasets. In all categories, our method or a baseline augmented with our POV conversion module outperforms previous state-of-the-art.

5.1.1 Effect of suggested path reranking

Our proposed path-reranking without POV conversion yields semi-extractive output summaries competitive with abstractive summarization baselines. Segmenting raw transcripts into topic groups with our method generally yields higher F-measures than using pre-segmented transcripts in semi-extractive summarization.

5.1.2 Effect of topic segmentation

Summarizing pre-segmented dialogue transcripts results in higher $R2$, while applying our topic segmentation method results in higher $R1$ and RL. This observation is in line with our method's emphasis on keyword extraction, in contrast to keyphrase extraction seen in several baselines (Boudin and Morin, 2013; Shang et al., 2018). Models that preserve *token adjacency* achieve higher $R2$, while models that preserve *token presence* achieve higher $R1$. RL additionally penalizes for wrong token order, but token order in extracted summaries tend to be well-preserved in word graph-based summarization schemes.

5.1.3 Effect of POV conversion module

Our POV conversion module improves benchmark scores on all tested baselines, as well as on our own system. It is only natural that a conversion module that translates text from semi-extractive to abstractive will raise scores on abstractive benchmarks. However, applying our POV module to *already abstractive* summarization systems resulted in higher scores in all cases. We attribute this to the fact that previous abstractive summarization systems do not generate sufficiently reportive summaries; past research either emphasize other linguistic aspects like hyponym conversion (Shang et al., 2018), or treat POV conversion as a byproduct of an end-to-end summarization pipeline (Fu et al., 2021).

5.2 Day-to-day, interview, screenplay, and debate summarization

Our method outperforms the LEAD-3 baseline on most benchmarks (Table 3). The model shows consistent performance across multiple domains in $R1$ and RL, but shows greater inconsistency in $R2$. Variance in the latter metric can be attributed, as in 5.1.2, to our model's tendency to optimize for single keywords rather than keyphrases. Robustness of our model, as measured by consistency of ROUGE measures across multiple datasets, is shown in Figure 4.

Notably, our method falters in the MediaSum benchmark. Compared to other benchmarks, MediaSum's reference summaries display heavy positional bias towards the beginning of its transcripts, which benefits the LEAD-3 approach. It also is the only dataset in which references summaries are

Transcript	Summary
Maya: Bring home the clothes that are hanging outside *Maya: All of them should be dry already and it looks like it's going to rain* *Boris: I'm not home right now* *Boris: I'll tell Brian to take care of that* *Maya: Fine, thanks*	*bring home the clothes that are hanging outside boris 'll tell brian to take care of that*
Keywords: 'care', 'clothes', 'home', 'thanks'	
Megan: Are we going to take a taxi to the opera? *Joseph: No, I'll take my car.* *Megan: Great, more convenient*	*are we going to take a taxi to the opera ?* *no , joseph 'll take my car*
Keywords: 'car', 'convenient', 'taxi', 'opera'	
Anne: You were right, he was lying to me :/ *Irene: Oh no, what happened?* *Jane: who?* *Jane: that Mark guy?* *Anne: yeah, he told me he's 30, today I saw his passport - he's 40* *Irene: You sure it's so important?* *Anne: he lied to me Irene*	*he lied to me he 's 30 , today anne saw his passport - he 's 40 yeah , he told me* *oh no , what happened? who ?* *annerene he lied to me : /*
Keywords: 'guy', '/', 'passport', 'yeah', 'today'	

Table 4: Summarizing the SAMSum corpus (Gliwa et al., 2019).

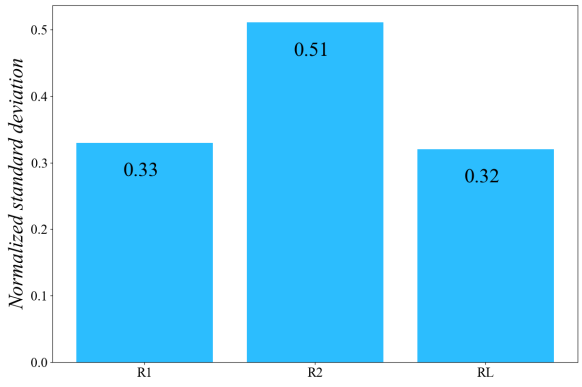

Figure 4: Normalized standard deviation (also called coefficient of variance) of R1, R2, and RL scores across all datasets. Normalized standard deviation is calculated as $\sigma/\bar{x}$, where σ is the standard deviation and $\bar{x}$ is the mean.

not generated for the purpose of summary evaluation, but are scraped from source news providers. Reference summaries for MediaSum utilize less reported speech compared to other datasets, and thus our POV module fails to boost the precision of summaries generated by our model.

6 Conclusion

6.1 Improving MSCG summarization

This paper improves upon previous work on multi-sentence compression graphs for summarization. We find that simpler and more adaptive path reranking schemes can boost summarization quality. We also demonstrate a promising possibility for integrating point-of-view conversion into summarization pipelines.

Compared to previous research, our model is still insufficient in keyphrase or bigram preservation. This phenomenon is captured by inconsistent $R2$ scores across benchmarks. We believe incorporating findings from keyphrase-based summarizers (Riedhammer et al., 2010; Boudin and Morin, 2013) can mitigate such shortcomings.

6.2 Avenues for future research

While our methods demonstrate improved benchmark results, its mostly heuristic nature leaves much room for enhancement through integration of statistical models. POV conversion in particular can benefit from deep learning-based approaches (Lee et al., 2020). With recent advances in unsupervised sequence to sequence transduction (Li et al.,

2020; He et al., 2020), we expect further research into more advanced POV conversion techniques will improve unsupervised dialogue summarization.

Another possibility to augment our research with deep learning is through employing graph networks (Cui et al., 2020) for representing MSCGs. With graph networks, each word node and edge can be represented as a contextualized vector. Such schemes will enable a more flexible and interpolatable manipulation of syntax captured by traditional word graphs.

One notable shortcoming of our system is the generation of summaries that lack grammatical coherence or fluency (Table 4). We intentionally leave out complex path filters that gauge linguistic validity or factual correctness. We only minimally inspect our summaries to check for inclusion of verb nodes, as in Filippova (2010). Our system can be easily augmented with such additional filters, which we leave for future work.

References

Florian Boudin and Emmanuel Morin. 2013. Keyphrase extraction for n-best reranking in multi-sentence compression. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4119–4135, Online. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2021a. Summscreen: A dataset for abstractive screenplay summarization. *arXiv preprint arXiv:2104.07091*.

Yulong Chen, Yang Liu, and Yue Zhang. 2021b. Dialogsum challenge: Summarizing real-life scenario dialogues. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 308–313.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5360–5371, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Fabbri, Faiaz Rahman, Imad Rizvi, Borui Wang, Haoran Li, Yashar Mehdad, and Dragomir Radev. 2021. ConvoSumm: Conversation summarization benchmark and improved abstractive summarization with argument mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6866–6880, Online. Association for Computational Linguistics.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 322–330, Beijing, China. Coling 2010 Organizing Committee.

Xiyan Fu, Yating Zhang, Tianyi Wang, Xiaozhong Liu, Changlong Sun, and Zhenglu Yang. 2021. Repsum: Unsupervised dialogue summarization based on replacement strategy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6042–6051.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348, Beijing, China. Coling 2010 Organizing Committee.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Matt Grenander, Yue Dong, Jackie Chi Kit Cheung, and Annie Louis. 2019. Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses. 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 6019–6024, Hong Kong, China. Association for Computational Linguistics.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. 2003. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Gunhee Lee, Vera Zu, Sai Srujana Buddi, Dennis Liang, Purva Kulkarni, and Jack FitzGerald. 2020. Converting the point of view of messages spoken to virtual assistants. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 154–163, Online. Association for Computational Linguistics.

Charles N Li. 2011. Direct speech and indirect speech: A functional study. In *Direct and indirect speech*, pages 29–46. De Gruyter Mouton.

Chunyuan Li, Xiang Gao, Yuan Li, Xiujun Li, Baolin Peng, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *EMNLP*.

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.

Wenfeng Liu, Yaling Gao, Jinming Li, and Yuzhen Yang. 2021. A combined extractive with abstractive model for summarization. *IEEE Access*, 9:43970–43980.

Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed H Awadallah, and Dragomir Radev. 2021. Dyle: Dynamic latent extraction for abstractive long-input summarization. *arXiv preprint arXiv:2110.08168*.

Iain McCowan, Jean Carletta, Wessel Kraaij, Simone Ashby, S Bourban, M Flynn, M Guillemot, Thomas Hain, J Kadlec, Vasilis Karaiskos, et al. 2005. The ami meeting corpus. In *Proceedings of the 5th international conference on methods and techniques in behavioral research*, volume 88, page 100. Citeseer.

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

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Barbara Partee. 1973. The syntax and semantics of quotation. In S. R. Anderson and P. Kiparsky, editors, *A Festschrift for Morris Halle*, pages 410–418. New York: Holt, Reinehart and Winston.

Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tür. 2010. Long story short–global unsupervised models for keyphrase based meeting summarization. *Speech Communication*, 52(10):801–815.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Jean-Pierre Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. *arXiv preprint arXiv:1805.05271*.

Antoine Tixier, Fragkiskos Malliaros, and Michalis Vazirgiannis. 2016. A graph degeneracy-based approach to keyword extraction. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1860–1870.

Antoine Tixier, Polykarpos Meladianos, and Michalis Vazirgiannis. 2017. Combining graph degeneracy and submodularity for unsupervised extractive summarization. In *Proceedings of the workshop on new frontiers in summarization*, pages 48–58.

Jin Y Yen. 1971. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716.

Xinyuan Zhang, Ruiyi Zhang, Manzil Zaheer, and Amr Ahmed. 2021. Unsupervised abstractive dialogue summarization for tete-a-tetes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14489–14497.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5927–5934, Online. Association for Computational Linguistics.

Yicheng Zou, Jun Lin, Lujun Zhao, Yangyang Kang, Zhuoren Jiang, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2021. Unsupervised summarization for chat logs with topic-oriented ranking and context-aware auto-encoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14674–14682.