A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal

Demian Gholipour Ghalandari[1,2], **Chris Hokamp**[1], **Nghia The Pham**[1],
John Glover[1], **Georgiana Ifrim**[2]
[1]Aylien Ltd., Dublin, Ireland
[2]Insight Centre for Data Analytics, University College Dublin, Ireland
[1]{first-name}@aylien.com
georgiana.ifrim@insight-centre.org

Abstract

Multi-document summarization (MDS) aims to compress the content in large document collections into short summaries and has important applications in story clustering for newsfeeds, presentation of search results, and timeline generation. However, there is a lack of datasets that realistically address such use cases at a scale large enough for training supervised models for this task. This work presents a new dataset for MDS that is large both in the total number of document clusters and in the size of individual clusters. We build this dataset by leveraging the Wikipedia Current Events Portal (WCEP), which provides concise and neutral human-written summaries of news events, with links to external source articles. We also automatically extend these source articles by looking for related articles in the Common Crawl archive. We provide a quantitative analysis of the dataset and empirical results for several state-of-the-art MDS techniques. The dataset is available at https://github.com/complementizer/wcep-mds-dataset.

1 Introduction

Text summarization has recently received increased attention with the rise of deep learning-based end-to-end models, both for extractive and abstractive variants. However, so far, only single-document summarization has profited from this trend. Multi-document summarization (MDS) still suffers from a lack of established large-scale datasets. This impedes the use of large deep learning models, which have greatly improved the state-of-the-art for various supervised NLP problems (Vaswani et al., 2017; Paulus et al., 2018; Devlin et al., 2019), and makes a robust evaluation difficult. Recently, several larger MDS datasets have been created: Zopf (2018); Liu et al. (2018); Fabbri et al. (2019). However, these datasets do not realistically resemble use cases with large automatically aggregated collections of news articles, focused on particular news events. This includes news event detection, news article search, and timeline generation. Given the prevalence of such applications, there is a pressing need for better datasets for these MDS use cases.

Human-written summary
Emperor Akihito abdicates the Chrysanthemum Throne in favor of his elder son, Crown Prince Naruhito. He is the first Emperor to abdicate in over two hundred years, since Emperor Kōkaku in 1817.
Headlines of source articles (WCEP)
• Defining the Heisei Era: Just how peaceful were the past 30 years?
• As a New Emperor Is Enthroned in Japan, His Wife Won't Be Allowed to Watch
Sample Headlines from Common Crawl
• Japanese Emperor Akihito to abdicate after three decades on throne
• Japan's Emperor Akihito says he is abdicating as of Tuesday at a ceremony, in his final official address to his people
• Akihito begins abdication rituals as Japan marks end of era

Table 1: Example event summary and linked source articles from the Wikipedia Current Events Portal, and additional extracted articles from Common Crawl.

In this paper, we present the Wikipedia Current Events Portal (WCEP) dataset, which is designed to address real-world MDS use cases. The dataset consists of 10,200 clusters with one human-written summary and 235 articles per cluster on average. We extract this dataset starting from the Wikipedia Current Events Portal (WCEP)[1]. Editors on WCEP write short summaries about news events and provide a small number of links to relevant source articles. We extract the summaries and source articles from WCEP and increase the number of source articles per summary by searching for similar articles in the Common Crawl News dataset[2]. As a result, we obtain large clusters of highly redundant news articles, resembling the output of news clustering applications. Table 1 shows an example of

[1]https://en.wikipedia.org/wiki/Portal:Current_events
[2]https://commoncrawl.org/2016/10/news-dataset-available/

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1302–1308

an event summary, with headlines from both the original article and from a sample of the associated additional sources. In our experiments, we test a range of unsupervised and supervised MDS methods to establish baseline results. We show that the additional articles lead to much higher upper bounds of performance for standard extractive summarization, and help to increase the performance of baseline MDS methods.

We summarize our contributions as follows:

- We present a new large-scale dataset for MDS, that is better aligned with several real-world industrial use cases.
- We provide an extensive analysis of the properties of this dataset.
- We provide empirical results for several baselines and state-of-the-art MDS methods aiming to facilitate future work on this dataset.

2 Related Work

2.1 Multi-Document Summarization

Extractive MDS models commonly focus on either ranking sentences by importance (Hong and Nenkova, 2014; Cao et al., 2015; Yasunaga et al., 2017) or on global optimization to find good combinations of sentences, using heuristic functions of summary quality (Gillick and Favre, 2009; Lin and Bilmes, 2011; Peyrard and Eckle-Kohler, 2016).

Several abstractive approaches for MDS are based on multi-sentence compression and sentence fusion (Ganesan et al., 2010; Banerjee et al., 2015; Chali et al., 2017; Nayeem et al., 2018). Recently, neural sequence-to-sequence models, which are the state-of-the-art for abstractive single-document summarization (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017), have been used for MDS, e.g., by applying them to extractive summaries (Liu et al., 2018) or by directly encoding multiple documents (Zhang et al., 2018; Fabbri et al., 2019).

2.2 Datasets for MDS

Datasets for MDS consist of clusters of source documents and at least one ground-truth summary assigned to each cluster. Commonly used traditional datasets include the **DUC 2004** (Paul and James, 2004) and **TAC 2011** (Owczarzak and Dang, 2011), which consist of only 50 and 100 document clusters with 10 news articles on average. The **MultiNews** dataset (Fabbri et al., 2019) is a recent large-scale MDS dataset, containing 56,000 clusters, but each cluster contains only 2.3 source documents on average. The sources were hand-picked by editors and do not reflect use cases with large automatically aggregated document collections. MultiNews has much more verbose summaries than WCEP.

Zopf (2018) created the **auto-*h*MDS** dataset by using the lead section of Wikipedia articles as summaries, and automatically searching for related documents on the web, resulting in 7,300 clusters. The **WikiSum** dataset (Liu et al., 2018) uses a similar approach and additionally uses cited sources on Wikipedia. The dataset contains 2.3 million clusters. These Wikipedia-based datasets also have long summaries about various topics, whereas our dataset focuses on short summaries about news events.

3 Dataset Construction

Wikipedia Current Events Portal: WCEP lists current news events on a daily basis. Each news event is presented as a summary with at least one link to external news articles. According to the editing guidelines[3], the summaries must be short, up to 30-40 words, and written in complete sentences in the present tense, avoiding opinions and sensationalism. Each event must be of international interest. Summaries are written in English, and news sources are preferably English.

Obtaining Articles Linked on WCEP: We parse the WCEP monthly pages to obtain a list of individual events, each with a list of URLs to external source articles. To prevent the source articles of the dataset from becoming unavailable over time, we use the 'Save Page Now' feature of the Internet Archive[4]. We request snapshots of all source articles that are not captured in the Internet Archive yet. We download and extract all articles from the Internet Archive Wayback Machine[5] using the newspaper3k[6] library.

Additional Source Articles: Each event from WCEP contains only 1.2 sources on average, meaning that most editors provide only one source article when they add a new event. In order to extend the set of input articles for each of the ground-truth

[3] https://en.wikipedia.org/wiki/Wikipedia:How_the_Current_events_page_works

[4] https://web.archive.org/save/

[5] https://archive.org/web/

[6] https://github.com/codelucas/newspaper

summaries, we search for similar articles in the Common Crawl News dataset[7].

We train a logistic regression classifier to decide whether to assign an article to a summary, using the original WCEP summaries and source articles as training data. For each event, we label the `article-summary` pair for each source article of the event as positive. We create negative examples by pairing each event with source articles from other events of the same date, resulting in a positive-negative ratio of 7:100. The features used by the classifier are listed in Table 2.

`tf-idf` similarity between title and summary
`tf-idf` similarity between body and summary
No. entities from summary appearing in title
No. linked entities from summary appearing in body

Table 2: Features used in the `article-summary` binary classifier.

We use unigram bag-of-words vectors with TF-IDF weighting and cosine similarity for the first two features. The entities are phrases in the WCEP summaries that the editors annotated with hyperlinks to other Wikipedia articles. We search for these entities in article titles and bodies by exact string matching. The classifier achieves 90% Precision and 74% Recall of positive examples on a hold-out set.

For each event in the original dataset, we apply the classifier to articles published in a window of ± 1 days of the event date and add those articles that pass a classification probability of 0.9. If an article is assigned to multiple events, we only add it to the event with the highest probability. This procedure increases the number of source articles per summary considerably (Table 4).

Final Dataset: Each example in the dataset consists of a ground-truth summary and a cluster of original source articles from WCEP, combined with additional articles from Common Crawl. The dataset has 10,200 clusters, which we split roughly into 80% training, 10% validation and 10% test (Table 3). The split is done chronologically, such that no event dates overlap between the splits. We also create a truncated version of the dataset with a maximum of 100 articles per cluster, by retaining all original articles and randomly sampling from the additional articles.

[7] https://commoncrawl.org/2016/10/news-dataset-available/

4 Dataset Statistics and Analysis

4.1 Overview

Table 3 shows the number of clusters and of articles from all clusters combined, for each dataset partition. Table 4 shows statistics for individual clusters. We show statistics for the entire dataset (WCEP-total), and for the truncated version (WCEP-100) used in our experiments. The high mean cluster size is mostly due to articles from Common Crawl.

	TRAIN	VAL	TEST	TOTAL
# clusters	8,158	1,020	1,022	10,200
# articles (WCEP-total)	1.67m	339k	373k	2.39m
# articles (WCEP-100)	494k	78k	78k	650k
period start	2016-8-25	2019-1-6	2019-5-8	-
period end	2019-1-5	2019-5-7	2019-8-20	-

Table 3: Size overview of the WCEP dataset.

	MIN	MAX	MEAN	MEDIAN
# articles (WCEP-total)	1	8411	234.5	78
# articles (WCEP-100)	1	100	63.7	78
# WCEP articles	1	5	1.2	1
# summary words	4	141	32	29
# summary sents	1	7	1.4	1

Table 4: Stats for individual clusters in WCEP dataset.

4.2 Quality of Additional Articles

To investigate how related the additional articles obtained from Common Crawl are to the summary they are assigned to, we randomly select 350 for manual annotation. We compare the article title and the first three sentences to the assigned summary, and pick one of the following three options: 1) "on-topic" if the article focuses on the event described in the summary, 2) "related" if the article mentions the event, but focuses on something else, e.g., follow-up, and 3) "unrelated" if there is no mention of the event. This results in 52% on-topic, 30% related, and 18% unrelated articles. We think that this amount of noise is acceptable, as it resembles noise present in applications with automatic content aggregation. Furthermore, summarization performance benefits from the additional articles in our experiments (see Section 5).

4.3 Extractive Strategies

Human-written summaries can vary in the degree of how extractive or abstractive they are, i.e., how much they copy or rephrase information in source documents. To quantify *extractiveness* in our

dataset, we use the measures *coverage* and *density* defined by Grusky et al. (2018):

$$Coverage(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} |f| \quad (1)$$

$$Density(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} |f|^2 \quad (2)$$

Given an article A consisting of tokens $\langle a_1, a_2, ..., a_n \rangle$ and its summary $S = \langle s_1, s_2, ..., s_n \rangle$, $F(A, S)$ is the set of token sequences (fragments) shared between A and S, identified in a greedy manner. Coverage measures the proportion of words from the summary appearing in these fragments. Density is related to the average length of shared fragments and measures how well a summary can be described as a series of extractions. In our case, A is the concatenation of all articles in a cluster.

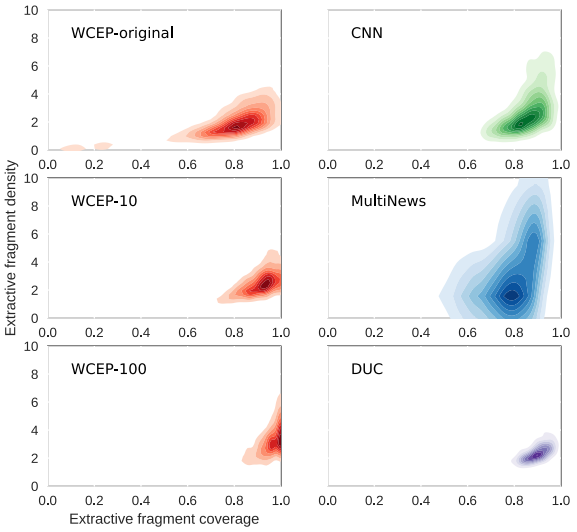

Figure 1: Coverage and density on different summarization datasets.

Figure 1 shows the distribution of coverage and density in different summarization datasets. WCEP-10 refers to a truncated version of our dataset with a maximum cluster size of 10. The WCEP dataset shows increased coverage if more articles from Common Crawl are added, i.e., all words of a summary tend to be present in larger clusters. High coverage suggests that retrieval and copy mechanisms within a cluster can be useful to generate summaries. Likely due to the short summary style and editor guidelines, high density, i.e., copying of long sequences, is not as common in WCEP as in the MultiNews dataset.

5 Experiments

5.1 Setup

Due to scalability issues of some of the tested methods, we use the truncated version of the dataset with a maximum of 100 articles per cluster (WCEP-100). The performance of the methods that we consider starts to plateau after 100 articles (see Figure 2). We set a maximum summary length of 40 tokens, which is in accordance with the editor guidelines in WCEP. This limit also corresponds to the optimal length of an extractive oracle optimizing ROUGE F1-scores[8]. We recommend to evaluate models with a dynamic (potentially longer) output length using F1-scores and optionally to provide Recall results with truncated summaries. Extractive methods should only return lists of full untruncated sentences up to that limit. We evaluate lowercased versions of summaries and do not modify ground-truth or system summaries otherwise. We compare and evaluate systems using F1-score and Recall of ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004). In the following, we abbreviate ROUGE-1 F1-score and Recall with R1-F and R1-R, etc.

5.2 Methods

We evaluate the following oracles and baselines to put evaluation scores into perspective:

- ORACLE (MULTI): Greedy oracle, adds sentences from a cluster that optimize R1-F of the constructed summary until R1-F decreases.
- ORACLE (SINGLE): Best of oracle summaries extracted from individual articles in a cluster.
- LEAD ORACLE: The lead (first sentences up to 40 words) of an individual article with the best R1-F score within a cluster.
- RANDOM LEAD: The lead of a randomly selected article, which is our alternative to the lead baseline used in single-document summarization.

We evaluate the unsupervised methods TEXTRANK (Mihalcea and Tarau, 2004), CENTROID (Radev et al., 2004) and SUBMODULAR (Lin and Bilmes, 2011). We test the following supervised methods:

- TSR: Regression-based sentence ranking using statistical features and averaged word embeddings (Ren et al., 2016).

[8]We tested lengths 25 to 50 in steps of 5. For these tests, the oracle is forced to pick a summary up to that length.

- BERTREG: Similar framework to TSR but with sentence embeddings computed by a pre-trained BERT model (Devlin et al., 2019). Refer to Appendix A.1 for more details.

We tune hyperparameters of the methods described above on the validation set of WCEP-100 (Appendix A.2). We also test a simple abstractive baseline, SUBMODULAR + ABS: We first create an extractive multi-document summary with a maximum of 100 words using SUBMODULAR. We pass this summary as a pseudo-article to the abstractive *bottom-up attention* model (Gehrmann et al., 2018) to generate the final summary. We use an implementation from OpenNMT[9] with a model pre-trained on the CNN/Daily Mail dataset. All tested methods apart from ORACLE (MULTI & SINGLE) observe the length limit of 40 tokens.

5.3 Results

Table 5 presents the results on the WCEP test set. The supervised methods TSR and BERTREG show advantages over unsupervised methods, but not by a large margin, which poses an interesting challenge for future work. The high extractive bounds defined by ORACLE (SINGLE) suggest that identifying important documents before summarization can be useful in this dataset. The dataset does not favor lead summaries: RANDOM LEAD is of low quality, and LEAD ORACLE has relatively low F-scores (although very high Recall). The SUBMODULAR + ABS heuristic for applying a pre-trained abstractive model does not perform well.

5.4 Effect of Additional Articles

Figure 2 shows how the performance of several methods on the test set increases with different amounts of additional articles from Common Crawl. Using 10 additional articles causes a steep improvement compared to only using the original source articles from WCEP. However, using more than 100 articles only leads to minimal gains.

6 Conclusion

We present a new large-scale MDS dataset for the news domain, consisting of large clusters of news articles, associated with short summaries about news events. We hope this dataset will facilitate the creation of real-world MDS systems for use cases such as summarizing news clusters or search results.

[9] https://opennmt.net/OpenNMT-py/Summarization.html

F-score			
Method	R1	R2	RL
ORACLE (MULTI)	0.558	0.29	0.4
ORACLE (SINGLE)	0.539	0.283	0.401
LEAD ORACLE	0.329	0.131	0.233
RANDOM LEAD	0.276	0.091	0.206
RANDOM	0.181	0.03	0.128
TEXTRANK	0.341	0.131	0.25
CENTROID	0.341	0.133	0.251
SUBMODULAR	0.344	0.131	0.25
TSR	**0.353**	**0.137**	**0.257**
BERTREG	0.35	0.135	0.255
SUBMODULAR+ABS	0.306	0.101	0.214
Recall			
Method	R1	R2	RL
ORACLE (MULTI)	0.645	0.331	0.458
ORACLE (SINGLE)	0.58	0.304	0.431
LEAD ORACLE	0.525	0.217	0.372
RANDOM LEAD	0.281	0.094	0.211
RANDOM	0.203	0.034	0.145
TEXTRANK	0.387	0.152	0.287
CENTROID	0.388	0.154	0.29
SUBMODULAR	0.393	0.15	0.289
TSR	**0.408**	**0.161**	**0.301**
BERTREG	0.407	0.16	**0.301**
SUBMODULAR+ABS	0.363	0.123	0.258

Table 5: Evaluation results on test set.

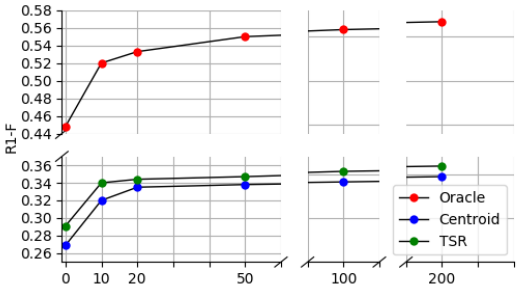

Figure 2: ROUGE-1 F1-scores for different numbers of supplementary articles from Common Crawl.

We conducted extensive experiments to establish baseline results, and we hope that future work on MDS will use this dataset as a benchmark. Important challenges for future work include how to scale deep learning methods to such large amounts of source documents and how to close the gap to the oracle methods.

Acknowledgments

This work was funded by the Irish Research Council (IRC) under grant number EBPPG/2018/23, the Science Foundation Ireland (SFI) under grant number 12/RC/2289_P2 and the enterprise partner Aylien Ltd.

References

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document abstractive summarization using ILP based multi-sentence compression. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 1208–1214.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Twenty-ninth AAAI conference on artificial intelligence*.

Yllias Chali, Moin Tanvee, and Mir Tafseer Nayeem. 2017. Towards Abstractive Multi-Document Summarization Using Submodular Function-Based Framework, Sentence Compression and Merging. *IJCNLP 2017* (2017), 418.

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)*. 4171–4186.

Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. 1074–1084. https://www.aclweb.org/anthology/P19-1102/

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)*. 340–348.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-Up Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4098–4109.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*. Association for Computational Linguistics, 10–18.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 708–719.

Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 712–721.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://www.aclweb.org/anthology/W04-1013

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 510–520.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by Summarizing Long Sequences. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Hyg0vbWC-

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 280–290.

Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2018. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1191–1204.

Karolina Owczarzak and Hoa Trang Dang. 2011. Overview of the TAC 2011 summarization track: Guided task and AESOP task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA, November*.

Over Paul and Yen James. 2004. An introduction to duc-2004. In *Proceedings of the 4th Document Understanding Conference (DUC 2004)*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A Deep Reinforced Model for Abstractive Summarization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HkAClQgA-

Maxime Peyrard and Judith Eckle-Kohler. 2016. A general optimization framework for multi-document summarization using genetic algorithms and swarm intelligence. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 247–257.

Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management* 40, 6 (2004), 919–938.

Pengjie Ren, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou. 2016. A Redundancy-Aware Sentence Regression Framework for Extractive Summarization. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, 33–43. https://www.aclweb.org/anthology/C16-1004

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 379–389.

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)*. 1073–1083.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based Neural Multi-Document Summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. 452–462.

Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. 2018. Towards a neural network approach to abstractive multi-document summarization. *arXiv preprint arXiv:1804.09010* (2018).

Markus Zopf. 2018. Auto-hMDS: Automatic Construction of a Large Heterogeneous Multilingual Multi-Document Summarization Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Markus Zopf, Eneldo Loza Mencía, and Johannes Fürnkranz. 2018. Which Scores to Predict in Sentence Regression for Text 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)*. 1782–1791.

A Appendices

A.1 BERTREG

This method uses a regression model to score and rank sentences. For a particular sentence, we obtain a contextualized embedding from a pre-trained BERT model[10]. We concatenate the embedding with several statistical and surface-form sentence features shown in Table 6.

length (in tokens)
position
stop word ratio
mean `tf`
mean `tf-idf`
mean `tf-icf`
mean `cluster-df`

Table 6: Features used for BERTREG apart from the contextual sentence embeddings.

The corpus-level document and cluster frequencies (cf) in `tf-idf` and `tf-icf` are obtained from the training set. `cluster-df` refers to the document frequency within a particular cluster. We feed this concatenated sentence vector to a feed-forward network with one hidden layer of size 256. The model is trained to predict the R1 F-score between a sentence and the summary of a cluster, using the mean squared error loss. We found the F-score to work better than Precision or Recall. We use the SGD optimizer, a learning rate of 0.02, and train for 8 epochs with batch size 8. To construct a summary, we predict scores using this model, rank sentences, and greedily pick sentences from the ranked list under a redundancy constraint, as used in TSR.

A.2 Implementation Details for Extractive Methods

We implement the methods TEXTRANK, CENTROID, TSR and BERTREG in a commonly used framework that greedily selects sentences from a ranked list while avoiding redundancy (Zopf et al., 2018). We measure `redundancy` as the proportion of bigrams in a new sentence that appear in an already selected sentence. For each method, we tune threshold values for redundancy from 0 to 1 in steps of 0.1. For SUBMODULAR, we tune a parameter called `diversity` with values 1 to 10 in steps of 1, which has a similar role as the `redundancy` threshold. We use 100 randomly selected clusters from the validation set in WCEP-100 for parameter tuning. We set a minimum sentence length of 7 tokens which avoids summaries slighly shorter than the 40 token limit to be padded with very short or broken sentences.

[10] We use the 12-layer model from https://github.com/hanxiao/bert-as-service