What Have We Achieved on Text Summarization?

Dandan Huang[1,2*], **Leyang Cui**[1,2,3*], **Sen Yang**[1,2*],
Guangsheng Bao[1,2], **Kun Wang**, **Jun Xie**[4], **Yue Zhang**[1,2†]
[1] School of Engineering, Westlake University
[2] Institute of Advanced Technology, Westlake Institute for Advanced Study
[3] Zhejiang University, [4] Tencent SPPD
{huangdandan, cuileyang, yangsen, baoguangsheng}@westlake.edu.cn,
wongkhun@outlook.com, stiffxie@tencent.com, yue.zhang@wias.org.cn

Abstract

Deep learning has led to significant improvement in text summarization with various methods investigated and improved ROUGE scores reported over the years. However, gaps still exist between summaries produced by automatic summarizers and human professionals. Aiming to gain more understanding of summarization systems with respect to their strengths and limits on a fine-grained syntactic and semantic level, we consult the Multidimensional Quality Metric[1] (MQM) and quantify 8 major sources of errors on 10 representative summarization models manually. Primarily, we find that 1) under similar settings, extractive summarizers are in general better than their abstractive counterparts thanks to strength in faithfulness and factual-consistency; 2) milestone techniques such as copy, coverage and hybrid extractive/abstractive methods do bring specific improvements but also demonstrate limitations; 3) pre-training techniques, and in particular sequence-to-sequence pre-training, are highly effective for improving text summarization, with BART giving the best results.

1 Introduction

Automatic text summarization has received constant research attention due to its practical importance. Existing methods can be categorized into extractive (Dorr et al., 2003; Mihalcea and Tarau, 2004; Nallapati et al., 2017) and abstractive (Jing and McKeown, 2000; Rush et al., 2015; See et al., 2017) methods, with the former directly selecting phrases and sentences from the original text as summaries, and the latter synthesizing an abridgment by using vocabulary words. Thanks to the resurgence of deep learning, neural architectures have been investigated for both extractive (Cheng and Lapata, 2016; Xu and Durrett, 2019) and abstractive (Nallapati et al., 2016; Lewis et al., 2019; Balachandran et al., 2020) summarization systems.

Although improved ROUGE scores have been reported on standard benchmarks such as Gigaword (Graff et al., 2003), NYT (Grusky et al., 2018) and CNN/DM (Hermann et al., 2015) over the years, it is commonly accepted that the quality of machine-generated summaries still falls far behind human written ones. As a part of the reason, ROUGE has been shown insufficient as a precise indicator on summarization quality evaluation (Liu and Liu, 2008; Böhm et al., 2019). In the research literature, human evaluation has been conducted as a complement (Narayan et al., 2018). However, human evaluation reports that accompany ROUGE scores are limited in scope and coverage. On a fine-grained level, it still remains uncertain what we have achieved overall and what fundamental changes each milestone technique has brought.

We aim to address the above issues by quantifying the primary sources of errors over representative models. In particular, following MQM (Mariana, 2014), we design 8 metrics on the *Accuracy* and *Fluency* aspects. Models are analysed by the overall error counts on a test set according to each metric, and therefore our evaluation can be more informative and objective compared with existing manual evaluation reports. We call this set of metrics **PolyTope**. Using PolyTope, we manually evaluate 10 text summarizers including Lead-3, TextRank (Mihalcea and Tarau, 2004), Sequence-to-sequence with Attention (Rush et al., 2015), SummaRuNNer (Nallapati et al., 2017), Point-Generator (See et al., 2017), Point-Generator-with-Coverage (Tu et al., 2016; See et al., 2017), Bottom-Up (Gehrmann et al., 2018), BertSumExt (Liu and Lapata, 2019), BertSumExtAbs (Liu and Lapata, 2019) and BART (Lewis et al., 2019), through

* Equal contribution.
† Corresponding author.
[1] MQM is a framework for declaring and describing human writing quality which stipulates a hierarchical listing of error types restricted to human writing and translation.

which we compare neural structures with traditional preneural ones, and abstractive models with their extractive counterparts, discussing the effectiveness of frequently-used techniques in summarization systems. Empirically, we find that:

1. Preneural vs Neural: Traditional rule-based methods are still strong baselines given powerful neural architectures.

2. Extractive vs Abstractive: Under similar settings, extractive approaches outperform abstractive models in general. The main shortcoming is *unnecessity* for extractive models, and *omission* / *intrinsic hallucination* for abstractive models.

3. Milestone Techniques: Copy works effectively in reproducing details. It also reduces duplication on the word level but tends to cause redundancy to a certain degree. Coverage solves repetition errors by a large margin, but shows limits in faithful content generation. Hybrid extractive/abstractive models reflect the relative strengths and weaknesses of the two methods.

4. Pre-training: Pre-training is highly effective for summarization, and even achieves a better content selection capability without copy and coverage mechanisms. Particularly, joint pre-training combining text understanding and generation gives the most salient advantage, with the BART model achieving by far the state-of-the-art results on both automatic and our human evaluations.

We release the test set, which includes 10 system outputs and their manually-labeled errors based on PolyTope, and a user-friendly evaluation toolkit to help future research both on evaluation methods and summarization systems[2].

2 Related Work

Extractive Summarization Early efforts based on statistical methods (Neto et al., 2002; Mihalcea and Tarau, 2004) make use of expertise knowledge to manually design features or rules. Recent work based on neural architectures considers summarization as a word or sentence level classification problem and addresses it by calculating sentence representations (Cheng and Lapata, 2016; Nallapati et al., 2017; Xu and Durrett, 2019). Zhong et al. (2020) adopts document-level features to rerank extractive summaries.

Abstractive Summarization Jing and McKeown (2000) presented a cut-paste based abstractive summarizer, which edits and merges extracted snippets into coherent sentences. Rush et al. (2015) proposed a seq2seq architecture for abstractive summarization. Subsequently, Transformer was used and outperformed traditional abstractive summarizer by ROUGE scores (Duan et al., 2019). Techniques such as AMR parsing (Liu et al., 2015), copy (Gu et al., 2016), coverage (Tu et al., 2016; See et al., 2017), smoothing (Müller et al., 2019) and pre-training (Lewis et al., 2019; Liu and Lapata, 2019) were also examined to enhance summarization. Hybrid abstractive and extractive methods adopt a two-step approach including content selection and text generation (Gehrmann et al., 2018; Hsu et al., 2018; Celikyilmaz et al., 2018), achieving higher performance than end-to-end models in ROUGE.

Analysis of Summarization There has been much work analyzing summarization systems based on ROUGE. Lapata and Barzilay (2005) explored the fundamental aspect of "coherence" in machine generated summaries. Zhang et al. (2018) analyzed abstractive systems, while Kedzie et al. (2018) and Zhong et al. (2019) searched for effective architectures in extractive summarization. Kryscinski et al. (2019) evaluated the overall quality of summarization in terms of redundancy, relevance and informativeness. All the above rely on automatic evaluation metrics. Our work is in line with these efforts in that we conduct a fine-grained evaluation on various aspects. Different from the above work, we use human evaluation instead of automatic evaluation. In fact, while yielding rich conclusions, the above analytical work has also exposed deficiencies of automatic toolkits. The quality of automatic evaluation is often criticized by the research community (Novikova et al., 2017; Zopf, 2018) for its insufficiency in neither permeating into the overall quality of generation-based texts (Liu and Liu, 2008) nor correlating with human judgements (Kryscinski et al., 2019).

There has also been analysis work augmenting ROUGE with human evaluation (Narayan et al., 2018; Liu and Lapata, 2019). Such work reports coarse-grained human evaluation scores which typ-

[2] https://github.com/hddbang/PolyTope

Methods / ROUGE	Extractive Methods				Abstractive Methods					
	Lead-3	**TextRank**	**Summa**	**BertExt**	**S2S**	**PG**	**PG-Coverage**	**Bottom-Up**	**BertAbs**	**BART**
ROUGE-1	39.20	40.20	39.60	43.25	31.33	36.44	39.53	41.22	42.13	**44.16**
ROUGE-2	15.70	17.56	16.20	20.24	11.81	15.66	17.28	18.68	19.60	**21.28**
ROUGE-L	35.50	36.44	35.30	39.63	28.80	33.42	36.38	38.34	39.18	**40.90**

Table 1: ROUGE scores of 10 summarizers on CNN/DM Dataset (non-anonymous version). We get the score of Lead-3 and TextRank from Nallapati et al. (2017) and Zhou et al. (2018), respectively.

ically consist of 2 to 3 aspects such as informativeness, fluency and succinctness. Recently, Maynez et al. (2020) conducted a human evaluation of 5 neural abstractive models on 500 articles. Their main goal is to verify the faithfulness and factuality in abstractive models. In contrast, we evaluate both rule-based baselines and extractive/abstractive summarizers on 8 error metrics, among which faithfulness and factuality are included.

Our work is also related to research on human evaluation for summarization. To this end, Pyramid (Nenkova and Passonneau, 2004) scores a summarizer based on its system output and multiple references. Annotators are requested to identify the smallest content units of semantic meaning, and then associate each unit with a weight by counting how many reference summaries contain this unit. The score of a summary is computed according to the number and weight of units. In addition to Pyramid, there are human evaluation metrics based on ranking (Narayan et al., 2018), best-worst scaling (Kiritchenko and Mohammad, 2017) and question answering (Clarke and Lapata, 2010). The above methods assign one score to each summarization output. In contrast to these methods, our error-count based metrics are motivated by MQM for human writing, and are more fine-grained and informative. We show more empirical contrast between evaluation metrics in Figure 3 in Section 6. Recently, Stiennon et al. (2020) uses human evaluation as a reward for training automatic summarizers, reporting significant improvement compared with models trained using reference summaries. Their work also demonstrates the usefulness of human evaluation in text summarization.

3 Models

We re-implement and evaluate 10 representative and influential methods. Their publicly reported ROUGE F1 scores are illustrated in Table 1.

3.1 Extractive Methods

Lead-3 Lead-3 is a commonly-used baseline, which simply selects the first 3 sentences as the summary. It is used as a standard baseline by most recent work (Cheng and Lapata, 2016; Gehrmann et al., 2018). Intuitively, the first 3 sentences of an article in news domain can likely be its abstract, so the results of Lead-3 can be a highly faithful approximation of human-written summary.

TextRank TextRank (Mihalcea and Tarau, 2004) is an unsupervised key text units selection method based on graph-based ranking models (Page et al., 1998). It defines "recommendation" by calculating co-similarity between sentences and yielding a weighted graph accordingly. Sentences with high weights are extracted as summaries. It is selected as a representative of statistical models.

SummaRuNNer SummaRuNNer (Nallapati et al., 2017) is a representative neural extractive model which selects full sentences from the input as a summary. It first encodes the input with a hierarchical BiGRU, then scans input sentences from left to right. An accumulated summary representation is generated by a weighted sum of all previous selections, which is fed into a logistic classifier to make the final prediction on summary.

BertSumExt BertSumExt (Liu and Lapata, 2019) takes pre-trained BERT (Devlin et al., 2019) as the sentence encoder and an additional Transformer as the document encoder. A classifier on sentence representation is used for sentence selection. It takes advantages of knowledge from fine-tuned BERT for generating better summaries.

3.2 Abstractive Methods

Seq2Seq with Attention The sequence-to-sequence model structure was first used for abstractive summarization by Rush et al. (2015). To allow effective and free text generation rather than simple selection and rearrangement, a target-to-source attention module is adopted to capture the information from every encoder hidden state. We follow the implementation of See et al. (2017).

Pointer-Generator See et al. (2017) introduces the pointer network (Vinyals et al., 2015) to address

Issue type	Sub Issue Type	Subject	Object	Predicate	Number&Time	Place&Name	Attribute	Function Word	Whole Sentence
Accuracy	Addition	Critical	Critical	Critical	Major	Major	Major	Minor	Major
	Omission	Critical	Critical	Critical	Critical	Major	Major	Minor	Critical
	Inacc Intrinsic	Critical	Critical	Critical	Critical	Critical	Major	Minor	N/A
	Inacc Extrinsic	Critical	Critical	Critical	Critical	Critical	Critical	Minor	N/A
	Pos Neg Aspect	N/A	N/A	Critical	N/A	N/A	Critical	N/A	N/A
Fluency	Word Order	N/A	N/A	Major	N/A	N/A	Major	Minor	N/A
	Word Form	Minor	Minor	Minor	Minor	Minor	Minor	Minor	N/A
	Duplication	Major	Major	Major	Major	Major	Major	Minor	Major

Table 2: PolyTope for summarization diagnostics. This error matrix avoids subjectivity as human judgers only need to annotate issue types and syntactic labels of each mistake. Severity rules and scores is predefined and automatically calculated, without providing their own preference and scores.

the problem that seq2seq models tend to reproduce factual details inaccurately. The method can both generate words from the vocabulary via a generator, and copy content from the source via a pointer.

Pointer-Generator-with-Coverage See et al. (2017) use the coverage mechanism (Tu et al., 2016) to avoid repetition problems. This mechanism calculates a coverage vector as an extra input for the attention mechanism to strengthen attention to different locations.

Bottom-Up Gehrmann et al. (2018) propose a two-step approach, first selecting potential output words and then generating a summary based on pointer-generator network. Bottom-Up is selected as a representative of hybrid models which integrate extractive and abstractive methods.

BertSumExtAbs BertSumExtAbs (Liu and Lapata, 2019) adopts the same encoder as BertSumExt, and a 6-layer Transformer decoder with randomly initialized parameters. It is selected as a representative of neural abstractive models with pretrained contextualized sentence representation.

BART Instead of pre-training the encoder only, BART (Lewis et al., 2019) jointly pre-trains a seq2seq model combining a bidirectional encoder and an auto-regressive decoder. Further fine-tuned on summarization datasets, it achieves the current state-of-the-art result using ROUGE.

4 Evaluation Method

We analyze system performance by using ROUGE (Lin, 2004) for automatic scoring and PolyTope for human scoring. ROUGE has been adopted by most work on summarization. It is a recall-based metric calculating lexical overlap between system output and human summaries. Particularly, ROUGE-1 is based on unigram overlaps, ROUGE-2 on bigrams and ROUGE-L on longest common subsequences.

PolyTope is an error-oriented fine-grained human evaluation method based on Multidimensional Quality Metric (MQM) (Mariana, 2014). In particular, it consists of 8 issue types (Section 4.1), 8 syntactic labels (Section 4.2) and a set of severity rules (Table 2) to locate errors and to automatically calculate an overall score for the tested document. As illustrated in Figure 3, compared with ROUGE, PolyTope is more fine-grained in offering detailed and diagnostic aspects of overall quality.

We develop an operating interface for annotation, which is shown in Appendix A.1. Particularly, a human annotator is presented the original text and an output summary in juxtaposition, and is asked to select segments that are deemed incorrect after reading. Upon a preliminary selection, he is asked to make a further selection among 8 issue types and 8 syntactic labels, respectively. An embedded severity score is then generated automatically for every incorrect segment, and the quality score is calculated for the annotated summary as:

$$\text{Score} = (1 - \frac{\sum_{\alpha \in I} \alpha * \text{Severity}_{\alpha}}{\text{word}_{\text{count}}}) * 100,$$

where $I \in \{\text{MINOR}, \text{MAJOR}, \text{CRITICAL}\}$, indicating the error count for each severity. Severity scores are deducted for errors of different severity, with the deduction ratio being set as 1:5:10 for MINOR, MAJOR and CRITICAL, respectively. $\text{word}_{\text{count}}$ is the total number of words in samples. For a skilled annotator, it takes 2.5-4 minutes averagely to complete annotation of one sample, of which 2-3 minutes are used for extensive reading and 0.5-1 minutes for annotation. After PolyTope evaluation, 3-dimensional error points show the overall quality of the tested model (Figure 1). The inter-annotator agreement over 20 documents is 0.8621 in terms of Pearson correlation coefficient, which shows that PolyTope can significantly reduce subjective bias of annotators. More human annotation details are illustrated in Appendix B.

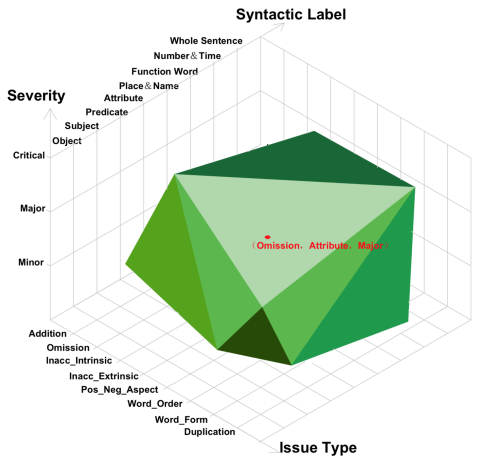


Figure 1: PolyTope verdicts each error by three coordinates according to its syntactic and semantic role.

4.1 Issue Type

Issue types of PolyTope can be categorized into *Accuracy* and *Fluency* issues, whose definitions can be traced to the MQM principle. *Accuracy*-related issues refer to the extent to which the content conveyed by the target summarization does not match or accurately reflect the source text. It comprises five sub-types:

Addition Unnecessary and irrelevant snippets from the source are included in the summary.

Omission Key point is missing from the output.

Inaccuracy Intrinsic Terms or concepts from the source are misrepresented and thus unfaithful.

Inaccuracy Extrinsic The summary has content not presented in the source and factually incorrect.

Positive-Negative Aspect The output summary represents positive statements whereas the source segment is negative, and vice versa.

Fluency issues refer to linguistic qualities of the text. Unlike *Accuracy*, *Fluency* is independent of the relationship between the source and the target. It comprises three sub-types:

Duplication A word or longer portion of the text is repeated unnecessarily.

Word Form Problems in the form of a word, including agreement, POS, tense-mood-aspect, etc.

Word Order Problems in the order of syntactic constituents of a sentence.

Their examples are elaborated in Appendix A.2.

4.2 Syntactic Label

Syntactic labels aim to locate errors, allowing tighter relevance between error issues and sentence constituents. According to ACE2005 (Automatic Content Extraction), we define 8 syntactic labels to distinguish sentence components, namely *Subject*, *Predicate*, *Object*, *Number&Time*, *Place&Name*, *Attribute*, *Function Word* and *Whole Sentence*. Their definitions are elaborated in Appendix A.3.

4.3 Severity

Severity is an indication of how severe a particular error is. It has three levels: MINOR, MAJOR, CRITICAL, calculated by the evaluation tool automatically given the human decision on the error type and syntactic label. In practice, each cell in Table 2 corresponds to a specific severity level. Issues with higher severity have more impact on perceived quality of the summary.

Minor Issues that do not impact usability or understandability of the content. For example, if grammar function word repeats itself, the redundant preposition is considered an error but does not render the text difficult to use or problematic.

Major Issues that impact usability or understandability of the content but do not render it unusable. For example, an additional attribute may result in extra effort for the reader to understand the intended meaning, but does not make the content unfit for purpose.

Critical Issues that render the content unfit for use. For example, an omitted subject that changes the meaning of the text would be considered critical. If the error prevents the reader from using the content as intended or if it presents incorrect information that could result in harm to the user, it must be categorized as critical. In general, even a single critical error is likely to cause serious problems.

5 Evaluating Model Performance

We evaluate the aforementioned 10 models using the above two metrics, focusing on comparisons between pre-neural and neural methods, extractive and abstractive methods, and better understanding the effects of milestone techniques such as copy, coverage, pre-training and hybrid abstractive/extractive models. We randomly sample 150 trials from the non-anonymized CNN/DM dataset (Hermann et al., 2015). When predicting

	Extractive Methods				Abstractive Methods					
	Lead-3	**TextRank**	**Summa**	**BertSumExt**	**S2S**	**PG**	**PG-Coverage**	**Bottom-Up**	**BertSumExtABS**	**BART**
ROUGE-1	41.63	33.81	41.11	42.69	31.87	38.89	39.90	41.19	41.87	43.28
ROUGE-2	19.62	13.71	20.15	21.19	13.07	19.64	19.00	19.98	21.02	21.28
ROUGE-L	35.55	26.47	36.40	35.95	29.48	35.92	35.01	36.52	34.16	38.13
ROUGE-1 Rank	**#4**	#9	#6	**#2**	#10	#8	#7	#5	**#3**	**#1**
Addition	329	272	156	160	125	117	143	207	165	135
Omission	196	309	193	185	329	286	256	287	213	115
Inacc_Intrinsic	0	0	0	0	304	14	16	68	7	2
Inacc_Extrinsic	0	0	0	0	42	0	0	4	0	0
Pos_Neg_Aspect	0	0	0	0	3	0	0	3	0	0
Word_Order	0	0	0	0	0	0	0	0	0	0
Word_Form	0	0	0	0	1	0	0	0	1	0
Duplication	17	12	36	9	139	68	11	6	3	2
Critical	192	302	191	184	588	284	257	333	213	112
Major	350	289	194	170	317	193	161	210	172	140
Minor	0	2	0	0	38	8	8	32	4	2
Errors / 1k Words	55	61	39	37	160	70	56	84	48	30
PolyTope Score	81.96	77.07	85.43	86.03	36.61	72.55	77.80	67.99	81.52	89.37
PolyTope Rank	**#4**	#7	**#3**	**#2**	#10	#8	#6	#9	#5	**#1**

Table 3: ROUGE and PolyTope results on 150 instances from CNN/DM dataset. ROUGE is the F1 score with stemming and stopwords not removed, giving the best agreement with human evaluation.

summaries, we select 3 sentences as the summary for extractive models following the original papers, and let the algorithms self-stop for abstractive models, which also give 3 sentences as the decoding result in most cases. Table 3 presents the performances based on PolyTope and ROUGE. Cases supporting observations below are illustrated in Appendix C.

5.1 Preneural vs Neural Models

On ROUGE-1, Lead-3 ranks the 2nd among extractive models, and the 4th among all models. On PolyTope, it ranks the 3rd among extractive models and the 4th among all models. This shows that the simple method stands as a strong baseline even among neural methods. TextRank ranks the 9th and 7th among all methods on ROUGE and PolyTope, respectively, still competitive to some abstractive neural models. On the negative side, these two methods show the largest numbers of *Addition* errors, which demonstrates that unsupervised methods are relatively weaker in filtering out useless information compared to the supervised methods.

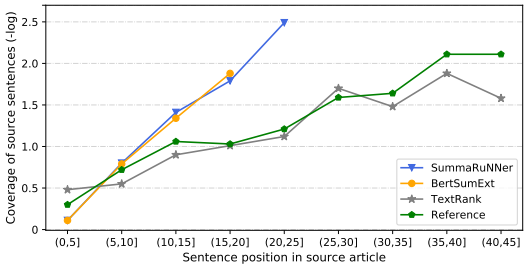
5.2 Extractive vs Abstractive Summarization

On ROUGE, there is no strong gap between extractive and abstractive methods, with BART and BertSumExt being the top abstractive and extractive models, respectively. On PolyTope, as representative of abstractive models, BART overwhelmingly outperforms the others ($p < 0.01$ using t-test). However, excluding BART, extractive models take the following top three places. Under similar settings, extractive methods are better ($p < 0.01$ using t-test) compared with abstractive counterparts (e.g. BertSumExt vs BertSumExtAbs, SummaRuNNer vs Point-Generator, Point-Generator-with-Coverage).

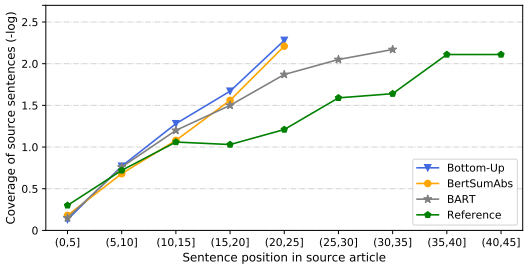
Extractive models tend to make only 3 types of errors, namely *Addition*, *Omission*, *Duplication*, while abstractive models make 4 to 7 types of errors. With respect to *Accuracy*, extractive methods are notably stronger in terms of *Inacc Intrinsic* and *Extrinsic*, which reflects that through directly copying snippets from the source, extractive methods are guaranteed to produce a summary with fair grammaticality, rationality and loyalty. However, extractive methods do not show stronger performances in *Addition* and *Omission*, which is because extracted sentences contain information not directly relevant to the main points. With regard to *Fluency*, two approaches are generally competitive with each other, showing that nowadays neural models are relatively effective in synthesizing coherent summaries.

5.3 Extractive Methods

We first compare neural methods BertSumExt and SummaRuNNer. BertSumExt gives better ROUGE-1/2 compared to SummaRuNNer, but the difference is not significant under ROUGE-L or PolyTope. Among detailed errors, BertSumExt demonstrates advantages only in *Duplication*, for the likely reason that the contextualized representations of the same phrases can be different by BERT encoding. It co-insides with previous findings (Kedzie et al., 2018) which demonstrate that more complicated architectures for producing sentence representations do not lead to better performance under the setting of extractive summarization. Given the fact that gold-standard extractive summaries are constructed according to ROUGE, the better ROUGE score of BertSumExt reflects the effectiveness of stronger representation on fitting training data.



(a) Extractive models.



(b) Abstractive models.

Figure 2: Distribution of source sentence used for content generation. X-axis: sentence position in source article. Y-axis: the negative log of coverage of sentence.

We then take statistical models into account. Figure 2a shows the distribution of source sentences used for content generation by each method. There is a high proportion in the first five sentences and a smooth tail over all positions for reference summaries. In contrast, BertSumExt and SummaRuNNer extract sentences mostly from the beginning, thereby missing useful information towards the end. TextRank improves the coverage slightly as it is graph-based and does not depend on sequence information. But as lack of supervision, the model has a large number of *Addition* and *Omission*.

5.4 Abstractive Methods

Copy The naïve seq2seq model suffers an *Inacc-Intrinsic* count of 304, the worst among all models compared. In contrast, the Point-Generator model reduces the error count to 14, demonstrating the effectiveness of the copy mechanism in faithfully reproducing details. Another interesting finding is that *Duplication* errors are also sharply reduced from 139 to 68, although the copy mechanism is not explicitly designed to address this problem. Further investigation shows that the reduced duplication patterns are mostly on the word level, while the effect on sentence-level duplication reduction is nearly zero. One likely reason is that the seq2seq decoder relies heavily on short-term history when deciding the next output word, without effective use of long-term dependencies. The Point-Generator model solves this problem by interpolating vocabulary level probability with copy probability, reducing reliance on previous outputs. On the negative side, the copy mechanism introduces *Addition* errors, because the auto-regressive point generator network tends to copy long sequences in entirety from the source, failing to interrupt copying at desirable length. This is also observed by Gehrmann et al. (2018) and Balachandran et al. (2020).

Coverage Coverage (Tu et al., 2016) is introduced to neural summarization systems to solve repetition issues. Compared with Point-Generator, Point-Generator-with-Coverage reduces *Duplication* errors from 68 to 11 and *Omission* errors from 286 to 256, proving that coverage is useful for better content selection. However, Point-Generator-with-Coverage yields more *Addition* and *Inacc_Intrinsic* errors than Point-Generator. We further extracted outputs of Point-Generator that do not have *Duplication* errors, finding that introducing the coverage mechanism reduces the average PolyTope scores from 77.54 to 74.07. It indicates that the coverage mechanism lacks inference capability and tends to generate summaries that incorrectly combine contents from the source into irrelevant information (Figure10 and Figure11 in Appendix C). This is likely because the coverage mechanism forces attention values from the decoder to the encoder to move monotonically to the right, and therefore can interfere with the original content selection process.

Hybrid Abstractive/Extractive Model Bottom-Up gives high ROUGE scores, but ranks the second *worst* on PolyTope. Compared with others, it suffers more from *Inaccuracy* errors. The inconsistency between ROUGE and PolyTope reflects the relative strengths and weaknesses of this method. On the positive side, it combines the advantages of extractive and abstractive models in selecting segments from the source and generating new contents in the summary, leading to a better recall. On the negative side, the abstractive generation model constrains copy attention only on the extracted snippets, thereby suffering from incomplete information sources for making inference and consequently lack of faithfulness and factual consistency.

Pre-training Both BertSumExtAbs and BART outperform the non-pretraining abstractive models by a large margin. They differ from the other methods in two aspects, namely the Transformer ar-

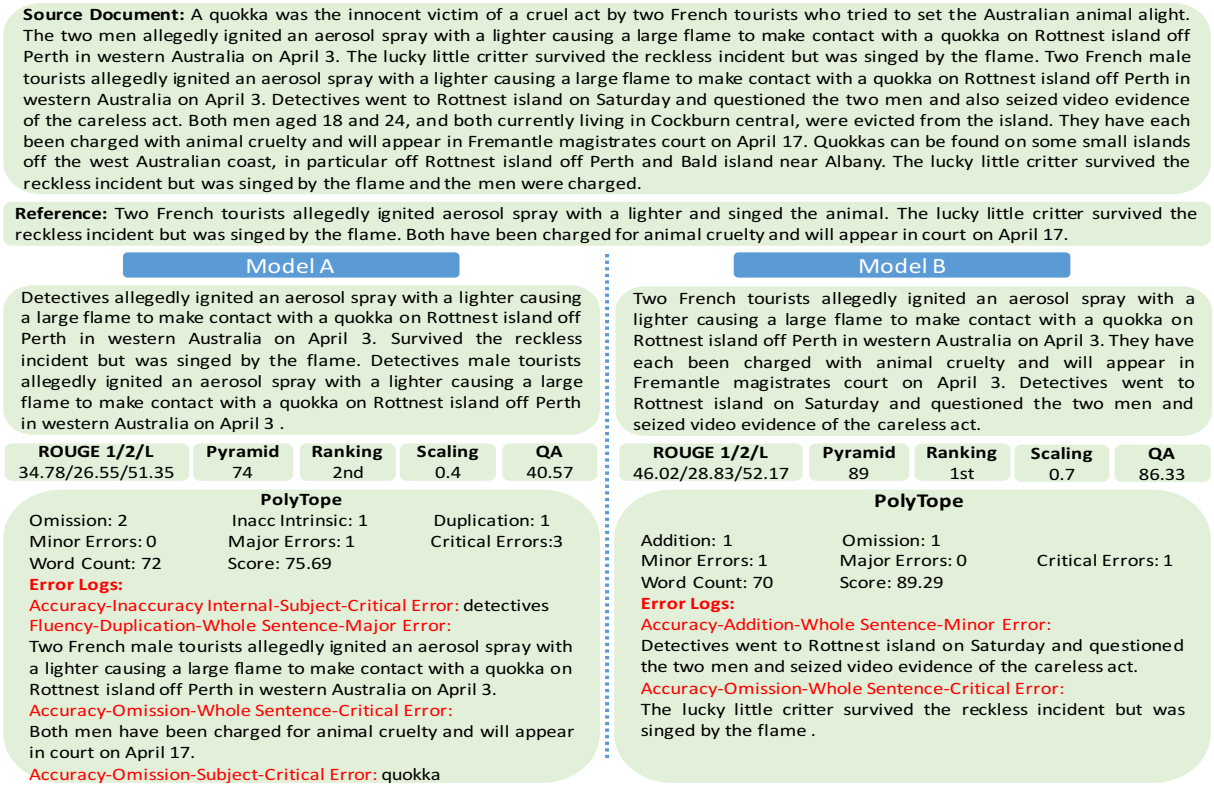


Figure 3: A case study that compares various evaluation methods with each other.

chitecture and contextualized knowledge. Since it has been shown that Transformer does not bring improved ROUGE compared with LSTM (Gehrmann et al., 2018; Zhong et al., 2019), knowledge encoded by large-scale pre-training is likely the key for their better performance. Without the help of copy and coverage, BertSumExtAbs gives less number of *Inacc* and *Duplication* errors, and BART further gives the least number in almost all errors, showing the strength of pre-training.

It is worth noting that BART ranks the 1st on both ROUGE and PolyTope among the 10 models. Different from BertSumExtAbs which pre-trains the encoder only, BART pre-trains the encoder and decoder jointly with seq2seq denoising auto-encoder tasks. It gives large improvements on *Addition*, *Omission* and *Inacc* errors, proving that unified pre-training for both understanding and generation is highly useful for content selection and combination. In particular, BART shows superior performance in handling the leading bias of CNN/DM dataset. Figure 2b shows the distribution of source sentences used for content generation by the abstractive methods. As can be seen, abstractive models tend to neglect sentences in the middle and at the end of source documents (e.g.,

		R-1	R-2	R-L
Instance	PolyTope	0.40	0.32	0.32
	Accuracy	0.31	0.26	0.25
	Fluency	0.07	0.41	0.01
System	PolyTope	0.78	0.73	0.52

Table 4: Pearson correlation coefficients between ROUGE scores and human annotations from the perspective of instance and system level, respectively.

Bottom-Up, BertSumExtAbs), indicating that performance of abstractive summarizers is strongly affected by the leading bias of dataset. In contrast, BART can attend to sentences all around the whole document, slightly closer to the distribution of golden reference. Intuitively, this improvement might result from the document rotation transformation of BART pre-training, which shuffles the sentences on the encoder side for the same decoder. We leave the verification to future work, which requires re-training of BART without document rotation transformation.

6 Analysis of Evaluation Methods

The main goal of this paper is to investigate the differences between summarization systems,

rather than to promote a human evaluation metric. Nonetheless, our dataset gives us a testbed to calculate the correlation between automatic and human evaluation methods. In this section, we report a contrast between ROUGE and PolyTope quantitatively, and between PolyTope and other human evaluation metrics qualitatively to demonstrate why we used PolyTope for our research goal.

First, research has shown that ROUGE is inconsistent with human evaluation for summary quality (Liu and Liu, 2008; Zopf, 2018; Kryscinski et al., 2019; Maynez et al., 2020). We evaluate ROUGE using PolyTope from the perspective of both instance-level and system-level performances. On the instance level, the individual 1500 outputs from the 10 models are adopted to calculate the Pearson correlation coefficients between ROUGE and PolyTope. Additionally, we select test instances that only make *Accuracy* or *Fluency* errors to better understand the correlation between ROUGE and *Accuracy*/*Fluency* aspects. On the system level, the overall scores of each model are adopted to calculate the Pearson correlation coefficients between ROUGE and PolyTope.

The results are summarized in Table 4. For the instance-level comparison, we find a weak correlation between ROUGE and human judgement. In addition, with respect to *Accuracy* and *Fluency*, ROUGE can measure *Accuracy* to a certain extent, and ROUGE-2 is better than ROUGE-1/L in terms of evaluating *Fluency*. For the system-level comparison, the Pearson correlation coefficient is 0.78, 0.73, 0.52 for ROUGE-1, ROUGE-2, and ROUGE-L, respectively, much higher than 0.40, 0.32, 0.32 on the instance level. This confirms that ROUGE is useful for ranking systems after aggregation of samples but is relatively weak for assessing single summary quality, where the fine grained PolyTope could help (Peyrard et al., 2017).

Second, Figure 3 shows results of two models on one test document by ROUGE, Pyramid, ranking, scaling, QA and PolyTope evaluation metrics. As can be seen from the figure, PolyTope offers more fine-grained information in quality evaluation. Sun et al. (2019) warned that human evaluation prefers to give higher scores to longer and more informative summaries. Under the setting of PolyTope, there was relatively little influence from the sentence length. Taking BertSumExt and BertSumExtAbs models as examples, the Pearson correlation coefficients between length of their outputs and the corresponding scores is 0.25 and 0.27, respectively, suggesting that PolyTope is more objective and meaningful for current models that produce summaries without pre-specified length.

Finally, we also evaluate the reference summaries of our 150 test trials by means of PolyTope, obtaining a general score of 96.41, with 63 errors in the *Accuracy* aspect and 0 errors in the *Fluency* aspect. Gold summaries did not receive full marks in the PolyTope evaluation, mainly because of hallucinating content. For example, a news article described an event as happening *"on Wednesday"* in a summary although the original document has *"on April 1"*. The human summary required external knowledge beyond the document and thus suffered penalization. Another common hallucination involved rhetorical but irrelevant sentences e.g., *"Click here for more news"*. In addition, there were a few grammatical issues that affect the accuracy. For example, in *"Piglet was born in China with only two front legs has learned to walk."*, there is a missing conjunction between two verb phrases.

7 Conclusion

We empirically compared 10 representative text summarizers using a fine-grained set of human evaluation metrics designed according to MQM for human writing, aiming to achieve a better understanding on neural text summarization systems and the effect of milestone techniques investigated recently. Our observations suggest that extractive summarizers generally outperform abstractive summarizers by human evaluation, and more details are also found about the unique advantages gained by copy, coverage, hybrid and especially pre-training technologies. The overall conclusions are largely in line with existing research, while we provide more details in an error diagnostics aspect.

Acknowledge

We thank all anonymous reviewers for their constructive comments. This work is supported by NSFC 61976180 and a research grant from Tencent Inc.

References

Vidhisha Balachandran, Artidoro Pagnoni, Jay Yoon Lee, Dheeraj Rajagopal, Jaime Carbonell, and Yulia Tsvetkov. 2020. Structsum: Incorporating latent and explicit sentence dependencies for single document summarization. *arXiv preprint arXiv:2003.00576*.

Florian Böhm, Yang Gao, Christian M. Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. 2019. Better rewards yield better summaries: Learning to summarise without references. 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 3108–3118, Hong Kong, China. Association for Computational Linguistics.

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

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

James Clarke and Mirella Lapata. 2010. Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 1–8.

Xiangyu Duan, Hoongfei Yu, Mingming Yin, Min Zhang, Weihua Luo, and Yue Zhang. 2019. Contrastive attention mechanism for abstractive sentence summarization. *CoRR*, abs/1910.13114.

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

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 708–719.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

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

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

Chris Kedzie, Kathleen R. 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, Brussels, Belgium, October 31 - November 4, 2018*, pages 1818–1828.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, volume 5, pages 1085–1090.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

Feifan Liu and Yang Liu. 2008. Correlation between ROUGE and human evaluation of extractive meeting summaries. In *Proceedings of ACL-08: HLT, Short Papers*, pages 201–204, Columbus, Ohio. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *CoRR*, abs/1908.08345.

Valerie Ruth Mariana. 2014. The multidimensional quality metric (mqm) framework: A new framework for translation quality assessment.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

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.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. 2019. When does label smoothing help? *arXiv preprint arXiv:1906.02629*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3075–3081.

Ramesh Nallapati, Bowen Zhou, Çağlar dos Santos, Cicero andxiang GuÌ‡lçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Joel Larocca Neto, Alex Alves Freitas, and Celso A. A. Kaestner. 2002. Automatic text summarization using a machine learning approach. In *Advances in Artificial Intelligence, 16th Brazilian Symposium on Artificial Intelligence, SBIA 2002, Porto de Galinhas/Recife, Brazil, November 11-14, 2002, Proceedings*, pages 205–215.

Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2241–2252.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia.

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*.

Simeng Sun, Ori Shapira, Ido Dagan, and Ani Nenkova. 2019. How to compare summarizers without target length? pitfalls, solutions and re-examination of the neural summarization literature. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 21–29, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. 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 3290–3301, Hong Kong, China. Association for Computational Linguistics.

Fangfang Zhang, Jin-ge Yao, and Rui Yan. 2018. On the abstractiveness of neural document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 785–790, Brussels, Belgium. Association for Computational Linguistics.

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.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.

Markus Zopf. 2018. Estimating summary quality with pairwise preferences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1687–1696, New Orleans, Louisiana. Association for Computational Linguistics.