Unsupervised Summarization with Customized Granularities

Ming Zhong[1], Yang Liu[2], Suyu Ge[1], Yuning Mao[1], Yizhu Jiao[1], Xingxing Zhang[3], Yichong Xu[2], Chenguang Zhu[2], Michael Zeng[2], Jiawei Han[1]
[1]University of Illinois at Urbana-Champaign
[2]Microsoft Cognitive Services Research Group
[3]Microsoft Research Asia
[1]{mingz5, hanj}@illinois.edu, [2]yaliu10@microsoft.com

ABSTRACT

Text summarization is a personalized and customized task, i.e., for one document, users often have different preferences for the summary. As a key aspect of customization in summarization, granularity is used to measure the semantic coverage between summary and source document. Coarse-grained summaries can only contain the most central event in the original text, while fine-grained summaries cover more sub-events and corresponding details. However, previous studies mostly develop systems in the single-granularity scenario. And models that can generate summaries with customizable semantic coverage still remain an under-explored topic. In this paper, we propose the first unsupervised multi-granularity summarization framework, GranuSum. We take events as the basic semantic units of the source documents and propose to rank these events by their salience. We also develop a model to summarize input documents with given events as anchors and hints. By inputting different numbers of events, GranuSum is capable of producing multi-granular summaries in an unsupervised manner. Meanwhile, to evaluate multi-granularity summarization models, we annotate a new benchmark *GranuDUC*, in which we write multiple summaries of different granularities for each document cluster. Experimental results confirm the substantial superiority of GranuSum on multi-granularity summarization over several baseline systems. Furthermore, by experimenting on conventional unsupervised abstractive summarization tasks, we find that GranuSum, by exploiting the event information, can also achieve new state-of-the-art results under this scenario, outperforming strong baselines.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**;
• **Information systems** → **Summarization**.

KEYWORDS

Text Summarization, Unsupervised Summarization, Granularity, Event

1 INTRODUCTION

In the information age, a plethora of information resources is at the fingertips of every user. Faced with a variety of complex and lengthy information, how to quickly understand their core idea has become a critical problem with increasing concerns. Therefore, the task of text summarization has grown in importance. Text summarization aims to condense and summarize long documents into a concise paragraph containing the essential points of the original texts. Notably, the requirements for summarization are highly customized and personalized for different users [13, 17, 26, 46]. Therefore, generating qualified summaries to meet different preferences should be a natural capability of summarization systems.

Granularity, a key aspect of customization in summarization, is used to measure the degree of semantic coverage between summary and source documents [34]. To cater to the diverse needs of readers, the granularity level of summaries often varies in a wide range. As shown in Table 1, given multiple news about Hurricane Mitch, the most compact summary (Coarse Granularity Level) accommodates only the most important event to help people grasp the overall

Table 1: An example from our multi-granularity summarization benchmark GranuDUC. Texts of the same color (blue, red) denote similar points described in different ways. Finer-grained summaries have higher semantic coverage with the original text.

Multiple News Articles about Hurricane Mitch
Honduras braced for potential catastrophe Tuesday as Hurricane Mitch roared through the northwest Caribbean, churning up high waves and intense rain ... (**Total 3,358 words**)
Summary of Coarse Granularity Level
Hurricane Mitch, category 5 hurricane, brought widespread death and destruction to Central American, and Honduras was especially hard hit. (**Total 19 words**)
Summary of Medium Granularity Level
Hurricane Mitch approached Honduras on Oct. 27, 1998 with winds up to 180mph a Category 5 storm ... The European Union, international relief agencies, Mexico, the U.S., Japan, Taiwan, the U.K. and U.N. sent financial aid, relief workers and supplies. (**Total 53 words**)
Summary of Fine Granularity Level
A category 5 storm, Hurricane Mitch roared across the northwest Caribbean with 180 mph winds across a 350-mile front ... The greatest losses were in Honduras where 6,076 people perished ... At least 569,000 people were homeless across Central America. Aid was sent from many sources (European Union, the UN, US and Mexico). The U.S. and European Union were joined by Pope John Paul II in a call for money and workers to help the stricken area. However, Relief efforts are hampered by extensive damage ... (**Total 133 words**)

Ming completed this work during his internship at Microsoft. Correspondence to: Yang Liu (yaliu10@microsoft.com).

picture of the input documents. Here, for instance, the location, specific level and consequences of the hurricane are included. Interested readers, on the other hand, may prefer more fine-grained summaries (Medium and Fine Granularity Level) to acquire additional details, such as how many casualties were caused and how different countries aided Honduras. Thus, multi-granularity summaries can meet the intent of different users and are more versatile in real-world applications.

However, most existing summarization models and benchmarks focus solely on single-granularity summarization, that is, they are only capable of generating summaries with similar semantic coverage. Single-granularity summarization limits the ability of these systems to adapt to different user preferences and generalize to a wider range of granularity scenarios. To alleviate this issue, some recent studies are dedicated to controlling the length of summary [17, 24, 32]. Given the unresolved redundancy issue of abstractive models [40], increasing the length of the summary may lead to a repetitive narration of the same event. Although these models can control the output length to some extent, they do not guarantee that the generated content can cover different numbers and the importance of events in the original text. So such models that do not take into account the semantic coverage of the summary and the input are not sufficient to act as a granularity-aware system. Another research direction is query-based or aspect-based summarization [18, 21, 56]. Based on different queries or aspect names, models can focus on different parts of the document and create summaries of various granularities. In practice, this requires a user to provide a query or aspect name, implying that the user must have some prior knowledge of the domain or topic of the source text.

From a data perspective, the lack of such *"single input with multiple summaries at different granularities"* data both limits the possibility of training multi-granularity summarization models with supervised learning methods and makes it infeasible to evaluate models' capability in this regard. As a consequence, the only relevant dataset currently available is Reddit [25], where each input post corresponds to two summaries of different lengths. The short summary is the title of the post, while the long summary is the TL;DR of the post. However, both versions have similar summary lengths (9.3 and 23.0 words on average) and tend to describe exactly the same event (via phrases and sentences, respectively), resulting in no granularity differences in content. It is insufficient to build and evaluate granularity-aware summarization models on this dataset. Hence constructing multi-granularity summarization systems and benchmarks is still an under-explored topic.

In this paper, we propose an unsupervised multi-granularity summarization framework called GranuSum. Unlike previous work based on supervised learning to provide guidance signals, such as salient sentences [14], keywords [22], and retrieved summaries [1], our approach does not rely on any manually labeled data. To measure the level of granularity, we first regard events as the basic semantic units of the input texts because events carry rich semantic information and are considered as informative representations in many NLP tasks [5, 28, 49]. Overall, our system consists of two event-related components: Event-aware Summarizer and Event Selector. Specifically, given the document and randomly selected events in it as hints, we pre-train an abstractive Summarizer that can recover event-related passages. Furthermore, in an unsupervised manner, our Event Selector selects the events with high salience from the original text by the following two steps: 1) Candidate events pruning: according to the relevance and redundancy scores, extract several important sentences from the document and treat the events in these sentences as a candidate set, and 2) Event ranking: rank and filter the event candidate set according to the degree of influence of each event on the final generated text. Finally, by selecting different numbers of anchor events based on Event Selector, we are able to control Summarizer to generate summaries containing different events, thus covering different numbers of semantic units of the original text. With this approach, the obtained GranuSum becomes an unsupervised framework with the ability of multi-granularity summary generation.

Considering that no dataset is qualified for evaluating multi-granularity summarization systems, we re-annotate DUC2004 [11] as the first benchmark in this direction (denoted as *GranuDUC*). Given multiple documents on the same topic, we annotate summaries at three levels of granularity with different coverage of the documents. Also, to utilize the existing datasets for a supplement evaluation, we propose to divide several large-scale summarization datasets into buckets with summaries at different granularity levels to further evaluate the model performance. Experimentally, GranuSum surpasses strong summarization systems on all the multi-granularity evaluations. Additionally, we conduct conventional unsupervised abstractive summarization experiments on three benchmarks in different domains. Experimental results demonstrate that benefiting from the event information and our unsupervised framework, GranuSum also substantially improves the previous state-of-the-art model under this traditional setting.

We summarize our contributions as follows:

- We propose the first multi-granularity summarization framework, GranuSum. Its ability to generate summaries with different semantic coverage in an unsupervised fashion allows it to cater to various users and to be applied to more practical scenarios.
- We develop the first testbed for multi-granularity summarization. The new benchmark we build can be used to evaluate the ability of models to generate summaries at different granularities. Moreover, we design an auxiliary bucket-based evaluation that can evaluate the quality of the generated summary in different granular scenarios.
- Our proposed model outperforms previous strong baselines in all the evaluation settings. In addition to the ability to generate multi-granular summaries, GranuSum achieves state-of-the-art results on three summarization benchmarks in news and scientific paper domains.

2 RELATED WORK

2.1 Text Summarization

Typically, there are two main paradigms to perform text summarization: *abstractive* [4, 8, 19, 35, 39] and *extractive* [7, 31, 36, 41, 54]. Abstractive approaches involve paraphrasing the corpus using novel words or sentences while extractive approaches generate summaries by selecting salient sentences from a document. Besides,

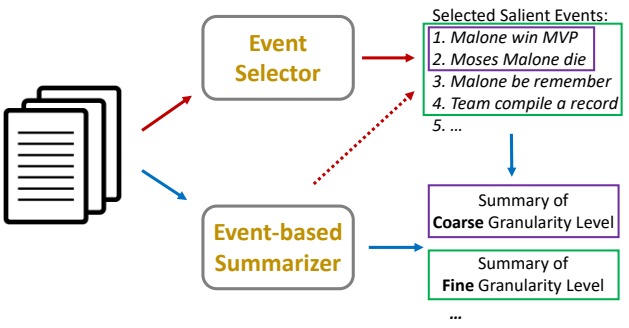

Figure 1: Overview of GranuSum. It consists of two components: Event Selector and Event-aware Summarizer. The red line indicates that Selector extracts the salient events from the original text, and the dotted line means that Summarizer assists in this process. The blue line denotes the multi-granularity summary generation process. By inputting different numbers of events as anchors (purple and green boxes), the Event-based Summarizer can generate summaries at different granularities.

some studies are a combination of these two methods. Pointer-Generator (PG) Network [40] can copy words directly from the source document during the decoding stage, which means it can extract from the original text or generate some novel words at the same time. Another perspective is to decompose the summarization task into two stages: extraction and generation [2, 33, 44]. For instance, Chen and Bansal [6] first select important sentences and then rewrite them by using PG to generate the final summary.

2.2 Customized Summarization

In order to meet the needs of different users, existing neural summarization systems attempt to control customization of the summary, such as the aspects of content [21, 56], summary length [24, 32] and writing style [1]. Also, some studies seek to accommodate multiple types of preferences simultaneously to achieve customized summarization. Fan et al. [17] additionally introduces different special marker tokens to the model to generate user-controllable summaries. He et al. [22] allows for entity-centric, length-controllable, and question-guided summarization by adjusting the prompts, i.e., changing the textual input in the form of a set of keywords or descriptive prompt words. However, these systems rely on supervised learning, but documents with multiple customized summaries are in short supply. Thus, we focus on unsupervised approaches and are committed to solving the granularity aspect, which remains an under-explored direction in customized summarization.

2.3 Unsupervised Summarization

In contrast to supervised learning, unsupervised models do not require any human-annotated summaries during training. Unsupervised summarization can also be divided into two branches: extractive methods and abstractive approaches. Most extractive methods rank the sentences and select the highest-ranked ones to form the summary. Specifically, they score sentences based on graph [15, 23, 38], centrality [29, 53], point-wise mutual information [37], or sentence-level self-attention in pre-trained models [45]. Another direction is unsupervised abstractive approaches, and these studies typically employ sequence-to-sequence auto-encoding method [9] with adversarial training and reinforcement learning [42]. In addition, Yang et al. [47] pre-train a Transformer model for unsupervised abstractive summarization by exploiting the lead bias phenomenon [40, 55] in the news domain. In this work, our framework is an unsupervised abstractive framework, and can be further enhanced on top of the extractive method.

Table 2: Five typical patterns and corresponding examples when we extract events (76 patterns in total). Here 'v' is a verb, 'n' stands for a noun, and 'a' denotes an adjective. All verbs remain in their original form. 'nsubj', 'dobj', 'xcomp', and 'nsubjpass' are syntactic relations.

Patterns	Examples
n_1-*nsubj*-v_1	Hurricane hit
n_1-*nsubj*-v_1-*dobj*-n_2	Hurricane damage buildings
n_1-*nsubj*-v_1-*xcomp*-*a*	People feel scared
n_1-*nsubj*-v_1-*xcomp*-v_2-*dobj*-n_2	Police want to save people
n_1-*nsubjpass*-v_1	Residents are injured

3 MULTI-GRANULARITY FRAMEWORK

In this section, we first describe in detail our framework GranuSum, which has two major components: Event-aware Summarizer and Event Selector. Combining them enables multi-granularity generation. The overall framework can be seen in Figure 1. Then, we introduce the new human-annotated benchmark, GranuDUC, which can be used for multi-granularity evaluation.

3.1 Event-Aware Summarizer

In this work, we focus on abstractive summarization approaches. The way we make the model perceive the granularity is by inputting hints with different degrees of specificity, and here we formalize the hints as a sequence of events.

Event Extraction. We follow previous work to define an event as a verb-centric phrase [49]. A lightweight method is utilized to extract events from open-domain unstructured data: we extract frequently-occurring syntactic patterns that contain verbs as events. On the basis of Zhang et al. [49], we extend a total of 76 syntactic patterns for matching events. Specifically, given a sentence s, we use a dependency parser to obtain its dependency parse tree and select all non-auxiliary verbs as centric tokens. Then, along the syntactic relationships between the selected verbs and other tokens, we extract the longest phrase that matches the designed patterns as events. As illustrated in Table 2, the most frequent pattern is n_1-nsubj-v_1, such as *Hurricane hit*. Another common pattern is n_1-nsubj-v_1-dobj-n_2, like *Hurricane damage buildings*. Here "nsubj" denotes an active relationship between nouns and verbs, while "nsubjpass" in another example represents a passive relationship between them. More detailed examples can be found in Table 3, we extract events from four selected sentences, and the colored text shows the locations of the events in the original document.

Table 3: Workflow of GranuSum and case studies. The colored text in Step 1 indicates the location of the extracted event in the original sentence. Events of the same color in Step 2 are redundant. Underlined text in Step 4 represents the overlapping content with the reference summary. Note that we pre-train an event-aware Summarizer before Step 1.

Step 1: Select Important Sentences based on Relevance and Redundancy Score, and Extract Events

- Malone was part of the 76ers' 1983 NBA championship team, and the club said he will forever be remembered as a genuine icon and pillar of the most storied era in the history of Philadelphia 76ers basketball. ⟶ club say | Malone be remember
- In the initial meeting in New York, Cunningham pulled Malone aside and let him know his expectations of the player who had won MVP honors in Houston the previous season by averaging 31.1 points and 14.7 rebounds. ⟶ Cunningham pull Malone | let hime know | win MVP | average 31.1 points and 14.7 rebounds
- In his first season with the Sixers, Malone won MVP awards by averaging 24.5 points and 15.3 rebounds during the regular season in which the team compiled a 65-17 record. ⟶ Malone win MVP | average 24.5 points and 15.3 rebounds | team compile a 65-17 record
- Moses Malone, a three-time NBA MVP and one of basketball's most ferocious rebounders, died Sunday, the Philadelphia 76ers said. ⟶ Moses Malone die | 76ers say

Step 2: Obtain a Candidate Set by Combining the Above Events

- Original Candidate Events: club say | Malone be remember | Cunningham pull Malone | let him know | win MVP | average 31.1 points and 14.7 rebounds | Malone win MVP | average 24.5 points and 15.3 rebounds | team compile a 65-17 record | Moses Malone die | 76ers say

Step 3: Event Ranking and Filtering (Event Selector)

- Ranked Candidate Events: Malone win MVP | Moses Malone die | Malone be remember | team compile a 65-17 record | Cunningham pull Malone | average 31.1 points and 14.7 rebounds | 76ers say | let him know

Step 4: Multi-Granularity Summary Generation (Event-based Summarizer)

- Coarse Granularity Level
  - Input: Malone win MVP | Moses Malone die ⟨seg⟩ ⟨mask⟩ Source News
  - Generated Summary: Moses Malone, a three-time NBA MVP and one of basketball's most ferocious rebounders, died on Sunday.
- Fine Granularity Level
  - Input: Malone win MVP | Moses Malone die | Malone be remember | team compile a 65-17 record ⟨seg⟩ ⟨mask⟩ Source News
  - Generated Summary: Moses Malone, a three-time NBA MVP and one of basketball's most ferocious rebounders, died on Sunday. He helped the team compile a 65-17 record in the first season. These achievements make him be remembered as a genuine icon and pillar in the history of 76ers basketball.

Summary Generated by PEGASUS

- Moses Malone, a three-time NBA MVP and one of basketball's most ferocious rebounders, died Sunday, the Philadelphia 76ers said. The 76ers issued a statement that said Malone had died. Malone was inducted into the Naismith Memorial Basketball Hall of Fame in 2001 and attended the induction ceremonies for the year's class in Springfield, Massachusetts this weekend.

Reference Summary

- Three-time NBA MVP and Philadelphia 76ers legend Moses Malone, who with Julius Erving in 1983 brought the City of Brotherly Love its first championship since 1967, has died at the age of 60, reports the Inquirer. Moses holds a special place in our hearts and will forever be remembered as a genuine icon and pillar of the most storied era in the history of Philadelphia 76ers basketball.

Event-based Summarizer Pre-training. Previous studies reveal that event information can be an effective building block for models to perform text generation [12, 20], so we attempt to obtain a Summarizer with the ability to generate event-related text in an unsupervised way. Concretely, we pre-train a sequence-to-sequence model in the following steps: 1) randomly select a few sentences from the text; 2) extract events in these selected sentences; 3) mask these sentences in the source document; 4) take events and masked text as input, and use these selected sentences as the target for the model. For the example in Table 1, for a paragraph of news as *"Honduras braced for potential catastrophe Tuesday. Hurricane Mitch roared through the Caribbean, churning up high waves and intense rain that sent coastal residents scurrying for safer ground. President declared a state of maximum alert and the Honduran military sent planes to pluck residents from their homes on islands near the coast"*, we 1) first randomly select a sentence: *"Hurricane Mitch roared through the Caribbean, churning up high waves and intense rain that sent coastal residents scurrying for safer ground"*, 2) extract events in it such as *Mitch roar*, *Mitch churn up wave and rain*, *send* and *resident scurry*, 3) then mask this sentence in the original paragraph, and finally 4) use extracted events and masked text as the input and regard the selected sentence as the target as follows:

- Input: Mitch roar | Mitch churn up wave and rain | send | resident scurry ⟨seg⟩ Honduras braced for potential catastrophe Tuesday. ⟨mask⟩ President declared a state of maximum alert

and the Honduran military sent planes to pluck residents from their homes on islands near the coast.
- Target: Hurricane Mitch roared through the Caribbean, churning up high waves and intense rain that sent coastal residents scurrying for safer ground.

where "|" token is used to split the different events, ⟨seg⟩ is the segmentation token and ⟨mask⟩ indicates that a sentence at this position is masked. In our experiments, we randomly mask 1 to n sentences from a document, which leads to n samples to pre-train our Summarizer. Here we set n to the smaller of a constant number 10 and one-third of the number of sentences in the document.

3.2 Event Selector

The salience of the selected events determines whether the Summarizer can generate a qualified summary or an irrelevant and uninformative paragraph. A long document can contain hundreds of events, and finding the best event subset involves an exponential search space. Therefore, it is crucial to have an Event Selector that selects the most important events in the text to feed to the Summarizer. Our event selector first reduces the search space by pruning out less salient events and sentences, and then ranks the remaining events using the pre-trained Summarizer.

Event Ranking. The salience of the different events extracted from the documents varies. Some of the events are informative and relevant to the original text, but others are too general or too specific. For instance, two events *club say* and *Malone be remember* can be extracted from the sentence *"The club said Malone will forever be remembered as a genuine icon and pillar in the Philadelphia 76ers team"*. The former is not important to this news about Malone, while the latter is indispensable. And in the sentence *"Malone won MVP awards by averaging 24.5 points and 15.3 rebounds"*, *"average 24.5 points and 15.3 rebounds"* is too detailed to be included in a high-level summary. Therefore, ranking candidate events is a key function of our Event Selector.

Inspired by Yuan et al. [48], where a pre-trained generative model is capable of evaluating the correlation between the input and the target, we also use our pre-trained Event-based Summarizer to calculate the salience score for each event. Given the candidate event set E and the source document D, our Summarizer can generate a candidate summary c_E. Whenever an event e in the input is removed, if the generated candidate summary $c_{E\setminus\{e\}}$ differs greatly from c_E, this indicates that the removed event e is salient. As in the example above, removing *"club say"* does not cause an obstacle for the model to recover the sentence whose main meaning is that Malone is remembered by people, while removing *"Malone be remember"* makes the model unable to output the correct sentence. Thus, the latter should be the more important event. Formally, the **Salience Score** of event e can be defined as:

$$\mathrm{Sal}(e) \stackrel{\text{def}}{=} -\mathrm{Sim}(c_{E\setminus\{e\}}; c_E), \tag{1}$$

$$\mathrm{Sim}(x_1, x_2) \stackrel{\text{def}}{=} \mathrm{R1}(\mathrm{x}_1, \mathrm{x}_2) + \mathrm{R2}(\mathrm{x}_1, \mathrm{x}_2), \tag{2}$$

where $\mathrm{Sim}(x_1, x_2)$ is a function based on ROUGE score [30] to measure the similarity between any two text sequences x_1 and x_2. R1 and R2 are ROUGE-1 and ROUGE-2 scores, respectively. We find that ROUGE-L and ROUGE-1 follow similar trends in our experiment, so we only include ROUGE-1 and ROUGE-2 for simplicity. Based on this score, our event Selector can rank all the events in the candidate set. However, a single sentence may contain multiple events, so a long document can encompass hundreds of events. Using all events as a candidate set would result in an unaffordable computational efficiency. To solve this issue, we prune the candidate events before we rank them.

Candidate Event Pruning. We expect to capture a small set of events that are relevant to the main topic while pruning redundant parts. Events with high relevance provide an efficient summary of the central points in the original text, while low redundancy ensures that the final summary is informative and concise. To this end, we first select several salient sentences and extract the events in them as a candidate set. For relevance, if a sentence has a high semantic overlap with other input sentences, it should have a higher centrality and a higher probability to be included in the summary [37]. Thus, we define the **Relevance Score** of each sentence as:

$$\mathrm{Rel}(s, D) \stackrel{\text{def}}{=} \mathrm{Sim}(s; D \setminus \{s\}), \tag{3}$$

where s means the sentence and D represents the given document. $D \setminus \{s\}$ indicates that the sentence s is removed from the original text D.

For redundancy, the sentences in the summary should contain low redundant information when compared with each other. So when extracting the k-th sentence, we define its **Redundancy Score** with respect to the previously selected sentences as follows:

$$\mathrm{Red}(s, S) \stackrel{\text{def}}{=} \sum_{i=1}^{k-1} \mathrm{Sim}(s_i; s), \tag{4}$$

where S is the previously selected summary containing k-1 sentences. By maximizing relevance and minimizing redundancy, we can calculate the **Importance Score** of each sentence as:

$$\mathrm{Imp(s)} = \lambda_1 \mathrm{Rel}(s, D) - \lambda_2 \mathrm{Red}(s, S). \tag{5}$$

Through iteratively calculating the score of each sentence, we can eventually obtain a fixed number of sentences and extract the events from them as a candidate set. At this point, candidate events usually account for less than 1/10 of all events in the original text, which greatly improves the efficiency of subsequent calculations.

As shown in Table 3, when we obtain candidate events from selected sentences, there are still different types of issues in the candidate set. Some generic and uninformative events, such as "*club say*" and "*let him know*", should have a lower priority for a summary. Although we introduce sentence-level redundancy score in the pruning step, as a finer-grained unit, events still suffer from redundancy problem (see events in Table 3 with the same color), e.g., both "win MVP", "Malone win MVP" and "average 31.1 points and 14.7 rebounds", "average 24.5 points and 15.3 rebounds" appear in the candidate set. However, after the events ranking and filter using our Event Selector, all of these issues are alleviated. In this case, our Selector regards "Malone win MVP", "Moses Malone die" and "Malone be remember" as the three most salient events, which is consistent with the original news. In addition, uninformative events ("*club say*" and "*let him know*") are ranked at the end of the candidate sets, and duplicate events ("win MVP" and "average 24.5

Table 4: Annotation of two samples in GranuDUC.

Sample 1: News about the Civil Suit against Microsoft
• **Summary of Coarse Granularity Level**: The Justice Department filed a civil suit against Microsoft to change its pattern of anti-competitive conduct on browser software.
• **Summary of Medium Granularity Level**: Business rivals have filed an anti-trust suit against Microsoft to break Microsoft Corp.'s monopoly on computer operating systems. The suit began with a Microsoft vs Netscape battle. The Government is examining Microsoft's financial records and painting a dark image of its Chairman Bill Gates. An unpublished book may be crucial to the trial.
• **Summary of Fine Granularity Level**: The Justice Department filed a suit against Microsoft for violation of the Sherman Act to change its anti-competitive conduct. The heart of the suit is the Internet browser battle between Microsoft and Netscape. Microsoft, it is argued, has told computer manufacturers that if they want Windows, they must forgo Netscape. Netscape complaint over browsers was central to the case, which grew to include Intel, IBM, Sun, Apple, AOL, and Intuit. The battle now extends far beyond that aiming at Microsoft's overall aggressive anti-competitive conduct. Microsoft's chairman, Bill Gates, usually seen as a visionary is portrayed in much darker tones in the trial. Microsoft was ordered to let Justice examine its records and sought a trial delay. An unpublished book provided evidence, which can be crucial to the trial.
Sample 2: News about the Health Condition of the Russian President
• **Summary of Coarse Granularity Level**: Russia President Boris Yeltsin's worsening heath condition caused great concern to the Russian leadership.
• **Summary of Medium Granularity Level**: During Russia President Boris Yeltsin's seven years in power, illness has often sidelined him. He recently cut short a trip to Central Asia because of a respiratory infection and he later canceled two out-of-country summits. Russia's leaders are calling for his resignation and question his legal right to seek reelection.
• **Summary of Fine Granularity Level**: Russia President Boris Yeltsin had a heart attack in 1996, followed by multiple bypass surgery. The cause of minor burns on his hand were not disclosed. On a trip to Uzbekistan he walked stiffly, stumbled, rambled and seemed confused. Ceremonies were canceled and the trip ended a day early. Yeltsin refuses to admit he is seriously ill and his condition is kept secret. He was treated with antibiotics and ordered to bed but went to the office anyway. Many Russians suspect he is sicker, question his ability to do his job, and want him to resign. The court was to judge on whether he could serve a third term, but he already has said he will not run.

points and 15.3 rebounds") are filtered out due to the lowest salience score. In general, the reasonable ranking of candidate events by the Selector plays a crucial role in improving the quality of subsequent multi-granularity summaries.

3.3 Multi-Granularity Summary Generation

With the Event-aware Summarizer and Event Selector, it is feasible to generate summaries at different granularities. By taking different numbers of ranked events as hints, Summarizer can sense the specific level of semantic coverage required to enable the generation of different summaries. In the inference phase, we follow the approach in [51] that no sentences are masked and the ⟨mask⟩ token is simply added at the beginning of source texts.

We can see from Table 3, to obtain the most condensed summary, the two most important events ("Malone win MVP" and "Moses Malone die") and the original news are fed to the model. Then, the pre-trained Summarizer can be aware of event-based cues and generate the corresponding sentence: "Moses Malone, a three-time NBA MVP and one of basketball's most ferocious rebounders, died on Sunday". As more events are input, our Summarizer also has the ability to adjust the order of the narrative to make the content more logical. In the summary of granularity level 2, the order in the prompt is "Malone be remember" then "team compile a 65-17 record", but the model first output "He helped the team compile a 65-17 record in the first season" and then "These achievements make him be remembered as a genuine icon and pillar in the history of 76ers basketball" to make the whole summary more coherent and intuitive. Compared to sentences selected from the source documents (see Step 1 in Table 3), the summary generated by GranuSum omits unimportant details and paraphrases to make it more concise. Abstractive models without guidance signals, such as PEGASUS, tend to generate some repetitive sentences (the first two sentences), and generate several less relevant sentences without capturing important events. In contrast, GranuSum can output summaries that are more relevant and faithful to the original text.

3.4 New Benchmark: GranuDUC

Considering that there is no qualified dataset for evaluating multi-granularity summarization models, we re-annotate a new benchmark called GranuDUC for this case on the basis of a multi-document summarization dataset DUC2004 [11]. Our annotation team consists of 5 PhD students in NLP or people with equivalent expertise. For each document cluster, annotators are required to read multiple source documents and write summaries at three different granularities. The annotators are informed to be aware that granularity is not distinguished by the number of sentences, but is defined by different semantic coverage of the original text. Specifically, we inform the annotators that "coarse granularity level" should include only the main event of the entire documents, "medium granularity level" should include several important conditions, results and processes surrounding the main topic, and "fine granularity level" should further include the details such as time and location for each sub-event. Summaries at different granularities require significantly different levels of semantic coverage. Newly annotated

Table 5: Statistics of all datasets we used in this paper. DUC2004 and GranuDUC are for testing only, and GranuDUC is annotated based on the source documents of DUC2004.

Datasets	# Samples	Len. of Doc.	Len. of Sum.
Multi-News	56K	1793	217
arXiV	214K	6021	272
DUC2004	50	5882	115
GranuDUC	50	5882	24 / 68 / 135

sentences are allowed to be copied or rewritten from DUC2004's original reference summaries. In addition, we require annotators not to use the same sentences in different summaries of a sample, even when describing the same event. Each annotated summary is required to be reviewed by another annotator, then these two people discuss and revise until an agreement is reached. In the end, GranuDUC contains a total of 50 clusters, each cluster contains an average of 10 related documents and 3 summaries of different granularity, ranging from 10 words to more than 200 words in length. To demonstrate the quality of GranuDUC, we include the annotations of two samples in Table 4.

4 EXPERIMENTS

We design three settings of experiments: 1) experiments on GranuDUC, 2) bucket-based evaluation and 3) unsupervised abstractive summarization. The first two settings constitute a new testbed for multi-granularity summarization. In addition to this scenario, the last experiment auxiliarily evaluates the quality of summaries generated by our framework under the conventional unsupervised abstractive summarization setting.

4.1 Experimental Setup

Datasets. To verify the effectiveness of our framework and to obtain more convincing results, we conduct experiments on four datasets from two domains. Notably, we focus on two types of datasets, multi-document and long-document summarization, which are two main scenarios where users call for a multi-granularity system. For multi-document summarization, we concatenate the multiple articles into a single text sequence and input it to the model. Besides our benchmark GranuDUC, we use the following three datasets. Detailed statistics are listed in Table 5.

Multi-News [16] is a large-scale multi-document summarization dataset in the news domain. We use it in bucket-based evaluation (Section 4.2.2) and unsupervised summarization experiments (Section 4.3).

DUC2004 [11] contains 50 clusters, each with 10 relevant news articles and 4 reference summaries written by humans. Due to its small size, it is usually used directly as a test set. We utilize it in the unsupervised summarization experiment (Section 4.3).

arXiv [10] is a collection of long documents derived from scientific papers. It takes the full text of the paper as input, and the corresponding abstract as the reference summary. We use it in the unsupervised summarization experiment (Section 4.3).

Implementation Details. To process long input text in Table 5, we choose the Longformer-Encoder-Decoder (LED) [3] equipped with sparse attention as our backbone model. For Multi-News and arXiv, we further pre-train LED with our event-related generation task on their training corpora (without using reference summaries) for a total of 10,000 and 30,000 steps, respectively. We set batch size to 32 and the maximum learning rate to 2e-5. λ_1 in the importance score is 1.0 and λ_2 is 0.4. Empirically, we extract 9 sentences for Multi-News and 4 sentences for arXiv to form a candidate set, and input 90% events according to salience score to the Summarizer under unsupervised summarization setting. For DUC2004 and GranuDUC, we test directly with the Summarizer pre-trained on Multi-News, since these datasets are all in the news domain. In all the experiments, we use standard pyrouge[1] to calculate ROUGE scores. Due to the limitation of computational resources, we truncate all input text to 3,072 tokens for LED models.

Baselines. We use the following baselines in this work:

BART [27] is the state-of-the-art sequence-to-sequence pre-trained model for various generation tasks, including abstractive dialogue generation, question answering, and text summarization. We use BART-large in all the experiments.

PEGASUS [50] is a powerful generation model with gap-sentences generation as a pretraining objective tailored for abstractive summarization. We use the large version of PEGASUS for comparison.

PEGASUS-event indicates that on top of PEGASUS, additional event information is input as a guiding signal to observe if it is helpful for PEGASUS.

LED [3] has the same architecture as BART, except that the attention in the encoder introduces additional local attention and extends the position embedding to 16K tokens by copying the original embedding. The parameters in the LED are initialized by the weights in BART.

LED-Length-Control (LED-LC) is a baseline that we obtained by further pre-training LED. Inspired by Fan et al. [17], given a document and the desired number of sentences k, we randomly place k sentences in the document with the ⟨mask⟩ token, and let the model recover these sentences. During inference, we input the text and the desired number of sentences as a hint to the model so that it can control the length of the output summary[2].

PRIMER [43] is a pre-trained model for multi-document summarization that reduces the need for dataset-specific architectures and extensive labeled data. It achieves state-of-the-art results on multi-document summarization datasets under multiple settings.

4.2 Multi-granularity Evaluation

The first testbed we built for multi-granularity summarization systems includes two evaluation methods: 1) To test the ability of the model to generate summaries with different granularity levels when given the same input, we evaluate different models on our benchmark GranuDUC; 2) To supplement the limited size of GranuDUC, we design a bucket-based evaluation approach, where we divide a large-scale summarization test set into different buckets based on

[1] pypi.python.org/pypi/pyrouge/0.1.3

[2] For example, if we need a two-sentence summary, the input format would be: ⟨2⟩ ⟨seg⟩ ⟨mask⟩ source documents. It is exactly the same as GranuSum in terms of the training details and data.

Table 6: Results on GranuDUC. The top half of the Table shows the result of the automatic metric ROUGE, and the bottom half presents the result of human evaluation, including fluency, relevance and faithfulness.

	Coarse Granularity Level			Medium Granularity Level			Fine Granularity Level		
Model	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
PEGASUS	20.74	4.20	15.11	24.86	4.39	14.34	29.79	5.70	14.83
PEGASUS-event	20.68	4.18	15.12	24.72	4.28	14.25	29.58	5.52	14.61
LED-LC	21.83	4.80	15.29	26.73	5.59	15.76	30.18	5.57	15.24
GranuSum	**23.61**	**6.60**	**17.12**	**29.69**	**6.84**	**16.23**	**34.71**	**7.49**	**17.42**
Model	**Flu.**	**Rel.**	**Faith.**	**Flu.**	**Rel.**	**Faith.**	**Flu.**	**Rel.**	**Faith.**
PEGASUS	3.25	3.36	3.15	3.46	3.49	2.72	3.73	3.44	2.58
LED-LC	3.97	3.39	3.08	3.93	3.57	3.14	3.67	3.62	2.73
GranuSum	**4.13**	**3.82**	**3.59**	**4.09**	**3.78**	**3.46**	**3.82**	**4.05**	**3.17**

Table 7: Result of bucket-based evaluation on Multi-news. We use BERTScore-recall to divide the test set into three buckets. Low means that the summary has low semantic coverage with the source documents. This approach can be used to evaluate the performance of the summarization system in scenarios with different granularity level.

Model	Low			Medium			High		
	R-1	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**
PRIMER	37.21	9.92	17.68	42.50	13.19	**20.24**	46.95	18.10	23.99
LED-LC	37.28	9.56	16.64	42.37	12.65	19.15	47.57	17.88	22.40
GranuSum	**38.19**	**10.27**	**18.07**	**44.73**	**14.12**	20.10	**50.23**	**19.62**	**24.11**
- Ranking	37.34	9.36	16.69	43.41	13.28	19.12	49.66	19.35	23.37

their granularity levels, and test the ability of models to generate qualified summaries in different granularity buckets.

4.2.1 Results on GranuDUC. The summaries of each sample in GranuDUC can be divided into three granularity levels, where coarse granularity level represents the most compact summary, and fine granularity level is the most fine-grained summary. We use automatic metrics ROUGE and perform the human evaluation to evaluate the performance of different models in GranuDUC. Notably, both LED-LC and GranuSum have the ability to adjust the output according to specific granularity scenarios. At three different granularity levels on GranuDUC, we let LED-LC output 1, 3 and 8 sentences which correspond to the average length of reference summaries at different granularities. For our model, we take the top 90% events with the highest salience score in the selected 1, 3, 8 sentences as the input hint.

Automatic Evaluation. As illustrated in Table 6, compared to PEGASUS, LED-LC can bring a certain degree of improvement due to the ability to control the length of the output summary. This improvement is not remarkable at fine granularity level. But for coarse and medium granularity levels, LED-LC can control the number of output sentences, while PEGASUS does not have a similar capability and it can only generate shorter summaries by truncating the output (to 32 and 64 words), which leads to performance degradation. On the other hand, GranuSum exceeds LED-LC and PEGASUS by a large margin in all the granularity levels. Although GranuSum and LED-LC are trained on the same data, GranuSum increases the R-1 score by 1.78 at coarse granularity level (21.83→23.61), and this improvement reaches to 4.53 at fine granularity level (30.18→34.71). With the benefit of event information as a guide, our model can generate more relevant and qualified summaries, and this advantage is more pronounced in fine-grained summaries. Therefore, GranuDUC is a more suitable system for multi-granularity scenarios than existing controllable summarization models.

Human Evaluation. In addition to the automatic metrics, we also conduct the human evaluation to have a more comprehensive understanding of the model output. Six graduate students are involved in this process to score the generated summaries from three different perspectives: fluency, relevance and faithfulness to the source documents. The score range is 1-5, with 1 being the worst and 5 the best. Each sample requires two people to discuss and agree on the scoring. According to the fluency scores in Table 6, both LED-LC and GranuDUC can generate coherent sentences, while PEGASUS performs poorly in coarse and medium granularity levels due to truncating the output to a fixed length. From the perspective of relevance and faithfulness, a clear trend is that the more fine-grained the summary, the more relevant it is to the original text and the more likely it is to contain factual errors. Specific to the models, since GranuSum has additional event-related information as hints, it does generate more relevant and faithful summaries in all granularity scenarios compared to other baselines.

4.2.2 Bucket-based Evaluation. Besides our benchmark, we seek to utilize existing large-scale datasets for multi-granularity evaluation. We first design a metric to calculate the granularity score

Table 8: Results of unsupervised abstractive summarization on three datasets.

Model	Multi-News			arXiv			DUC2004		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
LEAD	42.9	14.3	19.2	32.7	8.1	17.5	32.3	6.5	16.3
LED	17.3	3.7	10.4	15.0	3.1	10.8	16.6	3.0	12.0
BART	27.3	6.2	15.1	29.2	7.5	16.9	24.1	4.0	15.3
PEGASUS	32.0	10.1	16.7	29.5	7.9	17.1	32.7	7.4	17.6
PEGASUS-event	31.5	10.2	15.8	29.2	7.7	17.0	31.8	7.1	16.9
PRIMER	42.2	13.7	**20.6**	34.6	9.4	18.3	34.7	6.9	17.6
Selector	43.3	14.1	19.1	35.3	10.8	17.8	34.3	7.1	17.1
LED-LC	42.0	13.3	19.2	34.9	9.9	18.1	33.9	6.6	16.8
GranuSum	**43.7**	**14.2**	20.1	**36.0**	**11.3**	**18.6**	**34.8**	**7.3**	**17.9**
- Ranking	43.5	14.0	19.7	35.4	10.8	18.5	34.3	7.0	17.2

between the source document and the reference summary to categorize the different samples. Because the same events in original text and human-written summary may have different descriptions, we design a granularity score on the basis of BERTScore [52] to perform soft matching due to its ability to measure semantic coverage between two sequences. Specifically, we extract all the events in the source document and the reference summary as two event sequences, and calculate BERTScore-recall as the **Granularity Score** between them. Formally, it can be calculated as:

$$\text{GranuScore}(D, r) = \text{BERTScore-reall}(Event_D, Event_r), \quad (6)$$

where D is the source documents and r represents the reference summary. $Event_D$ denotes the event sequence concatenated by the events extracted from D. Intuitively, a high recall score of the reference summary to the original text indicates that it has high semantic coverage and therefore it is a summary at a high granularity level. According to this metric, we divide the samples in Multi-news test set into three buckets with exactly the same number of document clusters. Low indicates that the summary in this bucket has low semantic coverage with the source documents.

Although PRIMER is the state-of-the-art model, it does not have the flexibility to change the output in response to different buckets. For LED-LC, we let the model generate 7, 8, and 9 sentences in low, medium, and high buckets, respectively. For our model, we take the top 70%, 80%, and 90% of the events with the higher salience score (see Section 3.2) in 9 selected sentences as the input for three different buckets. As shown in Table 7, LED-LC has no significant benefits over PRIMER, indicating that controlling the output length and ignoring its connection to the original text is not a good solution for the multi-granularity system. In contrast, GranuSum achieves substantial improvements in all buckets compared to powerful baselines. In particular, in buckets with high semantic coverage, our model improves R-1 score by 3.28 compared to PRIMER. Besides, "- Ranking" means that we no longer filter out some events based on the salience score, which causes a performance drop. This confirms that our selector can indeed exclude irrelevant and redundant events and thus improve the quality of the generated summary.

4.3 Unsupervised Abstractive Summarization

The quality of the summary is a key factor for all summarization systems. So in addition to the multi-granularity scenario, we likewise compare GranuSum with conventional unsupervised abstractive summarization models. Table 8 provides results on three datasets. The first section includes a simple yet effective approach LEAD. It refers to extracting the first few sentences at the beginning of the text as a summary. LEAD is a strong baseline in the news domain because there is a lead bias problem [40, 55] in this field. The second section lists the performance of state-of-the-art summarization models and the last section contains the results of our models. Selector indicates that we extract several sentences from the source document based on our importance score described in Section 3.2 as the summary. GranuSum is our overall framework and "- Ranking" indicates that we remove the ranking and filtering step. Surprisingly, although GranuSum is not specially designed for the conventional unsupervised summarization task, it still beats all the competitors and achieves new state-of-the-art results on most metrics across datasets. Despite inputting the same hints, PEGASUS-event does not show the ability to exploit event information and even performs worse than PEGASUS. In contrast, our pre-trained Event-aware Summarizer incorporates event information well into the generated summaries and thus boosts performance. Furthermore, GranuSum outperforms Selector, which is a strong extractive baseline, and extractive approaches usually dominate unsupervised summarization tasks. We think this improvement is due to two reasons: 1) in the pre-training stage, important content in the masked sentences is easier to reconstruct due to the redundancy of input texts. Thus, our Summarizer learn to filter those unimportant content in inference, generating more concise summaries; 2) our Selector screens out less critical events which should not appear in the summary. Notably, our model improves the average 1.0 R-1 score on three datasets compared to the previous best results, which indicates that GranuSum is sufficient to generate qualified summaries besides its multi-granularity capability.

5 CONCLUSION

In this paper, we highlight the importance of multi-granularity summarization systems in catering to user preferences and applying

them to real-world scenarios. To facilitate research in this direction, we propose the first unsupervised multi-granularity summarization framework GranuSum and build a corresponding well-established testbed. Experiments in three different settings demonstrate the effectiveness of our framework.

REFERENCES

[1] Chenxin An, Ming Zhong, Zhichao Geng, Jianqiang Yang, and Xipeng Qiu. 2021. RetrievalSum: A Retrieval Enhanced Framework for Abstractive Summarization. *arXiv preprint arXiv:2109.07943* (2021).

[2] Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sang-goo Lee. 2019. Summary Level Training of Sentence Rewriting for Abstractive Summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. 10–20.

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[4] 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)*, Vol. 1. 1662–1675.

[5] Muhao Chen, Hongming Zhang, Qiang Ning, Manling Li, Heng Ji, Kathleen McKeown, and Dan Roth. 2021. Event-Centric Natural Language Processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Tutorial Abstracts*. 6–14.

[6] Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 675–686.

[7] 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 (Volume 1: Long Papers)*, Vol. 1. 484–494.

[8] Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 93–98.

[9] Eric Chu and Peter Liu. 2019. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*. PMLR, 1223–1232.

[10] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 615–621.

[11] Hoa Trang Dang. 2005. Overview of DUC 2005. In *Proceedings of the document understanding conference*, Vol. 2005. 1–12.

[12] Naomi Daniel, Dragomir Radev, and Timothy Allison. 2003. Sub-event based multi-document summarization. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*. 9–16.

[13] Alberto Díaz and Pablo Gervás. 2007. User-model based personalized summarization. *Information Processing & Management* 43, 6 (2007), 1715–1734.

[14] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A General Framework for Guided Neural Abstractive Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4830–4842.

[15] Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research* 22 (2004), 457–479.

[16] Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1074–1084.

[17] Angela Fan, David Grangier, and Michael Auli. 2018. Controllable Abstractive Summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. 45–54.

[18] Suyu Ge, Jiaxin Huang, Yu Meng, Sharon Wang, and Jiawei Han. 2021. Fine-Grained Opinion Summarization with Minimal Supervision. *arXiv preprint arXiv:2110.08845* (2021).

[19] 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*. 4098–4109.

[20] Goran Glavaš and Jan Šnajder. 2014. Event graphs for information retrieval and multi-document summarization. *Expert systems with applications* 41, 15 (2014), 6904–6916.

[21] Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. 2021. WikiAsp: A Dataset for Multi-domain Aspect-based Summarization. *Transactions of the Association for Computational Linguistics* 9 (2021), 211–225.

[22] Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. Ctrlsum: Towards generic controllable text summarization. *arXiv preprint arXiv:2012.04281* (2020).

[23] Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1515–1520.

[24] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling Output Length in Neural Encoder-Decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1328–1338.

[25] Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive Summarization of Reddit Posts with Multi-level Memory Networks. 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)*. 2519–2531.

[26] Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. 2009. Sentiment summarization: evaluating and learning user preferences. In *Proceedings of the 12th conference of the European chapter of the ACL (EACL 2009)*. 514–522.

[27] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 7871–7880.

[28] Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 684–695.

[29] Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Improving unsupervised extractive summarization with facet-aware modeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 1685–1697.

[30] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[31] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3721–3731.

[32] Yizhu Liu, Zhiyi Luo, and Kenny Zhu. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4110–4119.

[33] Alfonso Mendes, Shashi Narayan, Sebastião Miranda, Zita Marinho, André FT Martins, and Shay B Cohen. 2019. Jointly Extracting and Compressing Documents with Summary State Representations. 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)*. 3955–3966.

[34] Rutu Mulkar-Mehta, Jerry R Hobbs, and Eduard Hovy. 2011. Granularity in natural language discourse. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

[35] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Ça glar Gulçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. *CoNLL 2016* (2016), 280.

[36] 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)*, Vol. 1. 1747–1759.

[37] Vishakh Padmakumar and He He. 2021. Unsupervised Extractive Summarization using Pointwise Mutual Information. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 2505–2512.

[38] Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. 2015. Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 1949–1954.

[39] 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.

[40] 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)*, Vol. 1. 1073–1083.

[41] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuan-Jing Huang. 2020. Heterogeneous Graph Neural Networks for Extractive Document Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6209–6219.

[42] Yaushian Wang and Hung-Yi Lee. 2018. Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4187–4195.

[43] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2021. PRIMER: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization. *arXiv preprint arXiv:2110.08499* (2021).

[44] 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*. Association for Computational Linguistics, Hong Kong, China.

[45] Shusheng Xu, Xingxing Zhang, Yi Wu, Furu Wei, and Ming Zhou. 2020. Unsupervised Extractive Summarization by Pre-training Hierarchical Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. 1784–1795.

[46] Rui Yan, Jian-Yun Nie, and Xiaoming Li. 2011. Summarize what you are interested in: An optimization framework for interactive personalized summarization. In *Proceedings of the 2011 conference on empirical methods in natural language processing*. 1342–1351.

[47] Ziyi Yang, Chenguang Zhu, Robert Gmyr, Michael Zeng, Xuedong Huang, and Eric Darve. 2020. TED: A Pretrained Unsupervised Summarization Model with Theme Modeling and Denoising. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. 1865–1874.

[48] Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating Generated Text as Text Generation. *arXiv preprint arXiv:2106.11520* (2021).

[49] Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Wingki Leung. 2020. ASER: A Large-scale Eventuality Knowledge Graph. In *The Web Conference 2020-Proceedings of the World Wide Web Conference, WWW 2020*. 201.

[50] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*. PMLR, 11328–11339.

[51] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 11328–11339. http://proceedings.mlr.press/v119/zhang20ae.html

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

[53] Hao Zheng and Mirella Lapata. 2019. Sentence Centrality Revisited for Unsupervised Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 6236–6247.

[54] 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, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 6197–6208. https://doi.org/10.18653/v1/2020.acl-main.552

[55] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuan-Jing 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*. 1049–1058.

[56] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5905–5921.