Unsupervised Multi-Granularity Summarization

Anonymous ACL submission

Abstract

Text summarization is a user-preference based task. For one document, users often have different priorities for summary. Granularity level of the summary is a core component of these preferences. However, most existing studies focus solely on single-granularity scenarios, resulting in models that are limited to producing summaries with similar semantic coverage and are not customizable. In this paper, we propose the first unsupervised multi-granularity summarization framework, GranuSum. We regard events as basic semantic units of the original text and design a model that can take these events as anchors when generating summary. Meanwhile, by ranking these hint events and controlling the number of events, GranuSum is capable of generating summaries at different granularities in an unsupervised manner. We develop a testbed for the multi-granularity summarization task, including a new human-annotated benchmark GranuDUC where each document is paired with multiple summaries with different granularities. Extensive experiments on this benchmark and other large-scale datasets show that GranuSum substantially outperforms previous baselines. We also find that GranuSum exhibits impressive performance on conventional unsupervised abstractive summarization tasks via exploiting the event information, achieving new state-of-the-art results on three summarization datasets.

1 Introduction

In the information age, a plethora of information resources are at the fingertips of every user. Faced with a variety of complex and lengthy information, how to quickly understand the central idea has become a serious problem with increasing concerns. Therefore, the task of text summarization has grown in importance. Notably, the requirements for summarization are highly customized and personalized for different users (Díaz and Gervás, 2007; Lerman et al., 2009; Yan et al., 2011; Fan et al., 2018). Thus, generating qualified summaries to meet different preferences should be a natural capability of summarization systems.

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 Granularity Level 1
Hurricane Mitch, category 5 hurricane, brought widespread death and destruction to Central American, and Honduras was especially hard hit. (**Total 19 words**)
Summary of Granularity Level 2
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 Granularity Level 3
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**)

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.

Granularity, a key aspect of customization in summarization, is used to measure the degree of semantic coverage between summary and source documents (Mulkar-Mehta et al., 2011). To cater to the diverse needs of readers, the granularity level of summaries usually vary in a wide range. As shown in Table 1, given multiple news about Hurricane Mitch, the most compact summary (Granularity 1) can contain only the most important event to help people grasp the overall picture of the original text. Interested readers, on the other hand, may prefer more fine-grained summaries (Granularity 2 and 3) to acquire additional specifics, 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 studies and benchmarks focus on single-granularity summarization (they are only capable of generating summaries with similar semantic coverage). This limits the ability of these systems to adapt to different user preferences and generalize to a wider range of practical scenarios. To alleviate this issue, some recent works are dedicated to controlling the length of summary (Kikuchi et al., 2016; Fan et al., 2018; Liu et al., 2018). Although these models can control the length in certain degree, they do not take into account the level of semantic coverage between the summary and the original text. Another research direction is query-based or aspect-based summarization (Zhong et al., 2021; Hayashi et al., 2021; Ge et al., 2021). Based on different queries or aspect names, models can focus on the content of 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. Therefore, automatic granularity-aware summarization model 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 (Dou et al., 2021), keywords (He et al., 2020), and retrieved summaries (An et al., 2021), 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. Events carry rich semantic information and are considered as informative representations in many NLP tasks (Zhang et al., 2020a; Li et al., 2020; Chen et al., 2021). Inspired by this, 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 the hint, we pre-train a sequence-to-sequence Summarizer that can generate event-related passages. Furthermore, in an unsupervised manner, our Event Selector can select 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: by the degree of influence of each event on the target text generated by Summarizer, score and re-rank each candidate. Finally, by selecting different numbers of anchor events based on Event Selector, we are able to control Summarizer to generate summaries with different semantic coverage. With this pipeline, the obtained GranuSum becomes a powerful unsupervised system with the ability of multi-granularity summarization.

Considering that none of the existing datasets contain summaries of different granularities, we re-annotate DUC2004 (Dang, 2005) as the first benchmark for evaluating multi-granularity summarization systems. For multiple documents on the same topic, we annotate summaries at three levels of granularity with different coverage of the documents. We also use a bucket-based method to evaluate model performance in buckets with different semantic coverage levels. Experimentally, GranuSum surpasses strong baselines on all the multi-granularity evaluations. Furthermore, we conduct unsupervised abstractive summarization experiments on three mainstream datasets in different domains. Experimental results demonstrate that, benefiting from the event information, GranuSum substantially improves the previous state-of-the-art model under different settings.

2 Related Work

Customized Summarization In order to meet the needs of different users, existing neural summarization systems attempt to control different customizations of the summary, such as the aspects of content (Zhong et al., 2021; Hayashi et al., 2021), summary length (Kikuchi et al., 2016; Liu et al., 2018) and writing style (An et al., 2021). Also, some works seek to accommodate multiple types of preferences simultaneously to achieve customized summarization. Fan et al. (2018) additionally introduces different special marker tokens to the model to generate user-controllable summaries. He et al. (2020) 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, and diverse summary data 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.

Unsupervised Summarization In contrast to supervised learning, unsupervised models do not require any human-annotated summaries during training. Unsupervised summarization can 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 (Erkan and Radev, 2004; Hirao et al., 2013; Parveen et al., 2015), centrality (Zheng and Lapata, 2019; Liang et al., 2021), pointwise mutual information (Padmakumar and He, 2021), or sentence-level self-attention in pre-trained models (Xu et al., 2020). Another direction is unsupervised abstractive approaches, and these studies typically employ sequence-to-sequence auto-encoding method (Chu and Liu, 2019) with adversarial training and reinforcement learning (Wang and Lee, 2018). In addition, Yang et al. (2020) pre-train a Transformer model for unsupervised abstractive summarization by exploiting the lead bias phenomenon (See et al., 2017; Zhong et al., 2019) in the news domain. In this work, our framework is a combination of these two approaches, and can be further enhanced on top of the extractive method.

3 Multi-Granularity Framework

In this section, we describe in detail our framework GranuSum, which has two major components: Event-aware Summarizer and Event Selector. Combining them enables multi-granularity generation. Next, we introduce the new human-annotated benchmark, GranuDUC.

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 (Zhang et al., 2020a). 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. (2020a), we extend a total of 57 syntactic patterns for matching events. For instance, the most common patterns contain n_1-nsubj-v_1 (e.g., *Hurricane hits*) and n_1-nsubj-v_1-dobj-n_2 (e.g., *Earthquake damages buildings*)[1].

Event-based Summarizer Pre-training Previous studies reveal that event information can be an effective building block for models to generate summaries (Daniel et al., 2003; Glavaš and Šnajder, 2014), 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 target for the model. For example, for a dialogue text as *"Do you have any plans tomorrow? How about playing basketball? Sure, I just finished my homework, it's time to exercise."*, we can select *How about playing basketball?* and extract the event *play basketball*. In this case, the specific format given to the model is:

- Input: play basketball ⟨seg⟩ Do you have any plans tomorrow? ⟨mask⟩ Sure, I just finished my homework, it's time to exercise.
- Target: How about playing basketball?

where ⟨seg⟩ is 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 becomes 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

[1]Here nsubj and dobj are nominal subject and direct object, respectively. They are different relations between verbs and nouns.

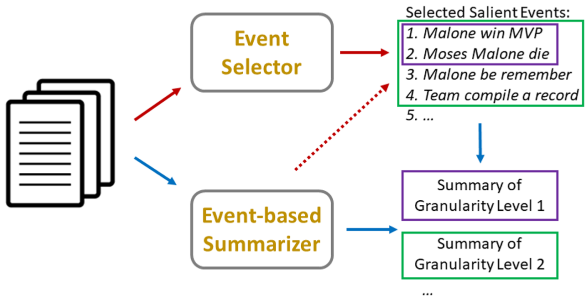

Figure 1: Overview of GRANUSUM. It consists of two components: Event Selector and Event-based Summarizer. The red line indicates that Selector extracts the salient events from the original text, and the dotted line means that Summarizer assists in this process. The blue line denotes the multi-granularity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

space by pruning out less salient event and sentences, and then ranks the remaining events using the pre-trained summarizer.

Event Ranking When we have several candidate events extracted from the source document, there are still differences in the salience of each event. Some of them 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. (2021), 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:

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

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

where $\mathrm{Sim}(x_1, x_2)$ is a function based on ROUGE score (Lin, 2004) 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. 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 a costly and unaffordable computational efficiency. To solve this issue, we prune the candidate events before we re-rank them.

Candidate Event Pruning We aim to collect a small set of candidate events from the given document, which can be considered as a compact summary of the original text. To this end, we first select several salient sentences and extract the events in them as a candidate set. Intuitively, if a sentence has a high semantic overlap with other input sentences, it will have a higher centrality and a higher probability to be included in the summary (Padmakumar and He, 2021). Thus, we define relevance score of each sentence as:

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

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.

In addition, the sentences in the summary should contain low redundancy information when compared with each other. When we extract the k-th sentence, we define its redundancy score with respect to the previous selected sentences as follows.

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

where S is the previously selected summary containing a total of 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.

3.3 Multi-Granularity Summary Generation

With the Event-aware Summarizer and Event Selector, it is possible 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. An example of our model output is as follows.

- Input 1: Malone win MVP | Moses Malone die ⟨seg⟩ ⟨mask⟩ [Source Documents]
- Summary of Granularity 1: Moses Malone, a three-time NBA MVP and one of basketball's most ferocious rebounders, died on Sunday.
- Input 2: Malone win MVP | Moses Malone die | Malone be remember | Team compile a 65-17 record ⟨seg⟩ ⟨mask⟩ [Source Documents]
- Summary of Granularity 2: 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 team.

In the inference phase, no sentences are masked and the ⟨mask⟩ token is simply added at the beginning of source texts, following (Zhang et al., 2020c). The example shows that events selected by our Selector are informative and highly relevant to Malone. When more events are added (*"Malone be remember"* and *"Team compile a 65-17 record"*), our Summarizer can output additional sentences that are relevant and faithful. In general, with an unsupervised framework, we are capable to generate qualified summaries at different granularities.

3.4 New Benchmark: GranuDUC

Considering that there is no dataset for evaluating multi-granularity summarization models, we re-annotate a new benchmark called GranuDUC for this case on the basis of multi-document dataset DUC2004 (Dang, 2005). Our annotation teams consists of 4 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 summary of granularity level 1 is limited to 1 sentence, the summary of granularity level 2 should be 3-5 sentences, and the summary of granularity level 3 contains 7-10 sentences. Newly annotated sentences are allowed to be copied or rewritten from DUC2004's original reference summaries. In addition, we required 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 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.

4 Experiments

To evaluate our model, 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. Respectively, they are employed to evaluate the ability of a model to generate multi-granularity summaries and the model performance on samples of different semantic coverage. In addition to multi-granularity scenarios, the last experiment auxiliarily evaluates the quality of summaries generated by our framework under 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 and input it to the model. Besides our benchmark GranuDUC, we use the following three datasets.

Multi-News (Fabbri et al., 2019) 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 (Dang, 2005) contains 50 clusters, each with 10 relevant news articles and 4 reference summaries written by human. Due to its small size, it is used directly as a test set. We use it in the unsupervised summarization experiment (Section 4.3).

ArXiv (Cohan et al., 2018) 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, we choose the Longformer-Encoder-Decoder (LED) (Beltagy et al., 2020) 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 the training corpus (without using reference summaries) for total 10,000 and 30,000 steps, respectively. The first 10% of these are warm-up steps. 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 Summaizer pre-trained on Multi-News, since these datasets are all in the news domain. In all the experiments, we use standard pyrouge[2] to calulate ROUGE scores. Due to the limitation of computational resources, we truncate all input text to 3,072 tokens for LED models.

Baselines We compare GranuSum with strong baselines as follows:

BART (Lewis et al., 2020) is the state-of-the-art sequence-to-sequence pre-trained model for various generation tasks, including abstractive dialogue, question answering, and text summarization. We use BART-large in all the experiments.

PEGASUS (Zhang et al., 2020b) is a powerful generation model with gap-sentences generation as a pretraining objective tailored for abstractive text summarization. We use the large version of PEGASUS for comparison.

LED (Beltagy et al., 2020) has the same architecture as BART, except that the attention in 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.

PRIMER (Xiao et al., 2021) is a pre-trained model for multi-document summarization that reduces the need for dataset-specific architectures and extensive labeled data. It achieved state-of-the-art results on multi-document summarizaion datasets under multiple settings.

LED-Length-Control (LED-LC) is a baseline that we obtained by further pre-training LED. Inspired by Fan et al. (2018). Given a document and the desired number of sentences k, we randomly place k sentences in the document with the $\langle$mask$\rangle$ token, and let the model to 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 length of the output summary. For example, if we need a two-sentence summary, the input format would be: $\langle$2$\rangle$ $\langle$seg$\rangle$ $\langle$mask$\rangle$ source documents. It is exactly the same as GranuSum in terms of the training details and data.

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 level when given the same document, we evaluate different models on our proposed 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 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 granularity level 1 represents the most compact summary, and granularity level 3 is the most fine-grained summary. We use automatic metrics ROUGE and perform 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, respectively. For our model, we first extract 1, 3, and 8 sentences based on impor-

[2] 4pypi.python.org/pypi/pyrouge/0.1.3

	Granularity 1			Granularity 2			Granularity 3		
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
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 2: 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.

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

Table 3: 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.

tance score, and then select the top 90% events with the highest salience score as the input hint.

Automatic Evaluation As illustrated in Table 2, 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 granularity level 3. But for granularity levels 1 and 2, 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 a 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 granularity level 1 (21.83→23.61), and this improvement reaches to 4.53 at granularity 3 (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 human evaluation to have a more comprehensive understanding of the model output. A total of 6 graduate students are involved in this evaluation 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 being the best. Each sample requires two people to discuss and agree on the scoring. According to the fluency scores in Table 2, both LED-LC and GRANUDUC can generate coherent sentences, while PEGASUS performs poorly in granularity levels 1 and 2 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 Approach

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 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 use BERTScore (Zhang et al., 2019) 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 text

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
RULE	43.3	14.1	19.1	35.3	10.8	17.8	34.3	7.1	17.1
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
PRIMER	42.2	13.7	**20.6**	34.6	9.4	18.3	34.7	6.9	17.6
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

Table 4: Results of unsupervised abstractive summarization on three datasets.

sequences, and calculate BERTScore-recall as the granularity score between them. Based on 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 first extract 9 sentences, and then take the top 70%, 80%, and 90% of the events with the higher salience score (see Section 3.2) in these sentences as the input for three different buckets. As shown in Table 3, 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 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 the 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 events and thus improve the quality of the generated summary.

4.3 Unsupervsied Abstractive Summarization

The quality of the summary is a key factor for all summarization systems. So despite the multi-granularity scenario, we likewise compare GRANUSUM with unsupervised abstractive summarization models. Table 4 provides results on three datasets. The first section includes two baselines: LEAD and RULE. LEAD is a strong baseline in the news domain because there is a lead bias problem (See et al., 2017; Zhong et al., 2019) in this field. It refers to extracting the first few sentences at the beginning of the text as a summary. RULE indicates that we extract several sentences from the source document based on our importance score described in Section 3.2 as the summary. The second section lists the performance of state-of-the-art summarization models and the last section contains the results of our model.

Surprisingly, although GRANUSUM is not specially designed for the conventional unsupervised summarization task, when enhanced with event-based information, it beats all the competitors under this setting and achieves new state-of-the-art results on most metrics across datasets. Notably, GRANUSUM outperforms RULE, which is a strong extractive baseline, and extractive approaches usually dominate unsupervised summarization tasks. We believe this improvement is due to two reasons: 1) In pre-training, important content in the masked sentences are 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. In addition, our model can boost average 1.0 R-1 score on three datasets compared to the previous best results. This indicates that our model 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

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*.

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

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*, pages 6–14.

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

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)*, pages 615–621.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12.

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

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

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

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

Alexander 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*, pages 1074–1084.

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

Suyu Ge, Jiaxin Huang, Yu Meng, Sharon Wang, and Jiawei Han. 2021. Fine-grained opinion summarization with minimal supervision. *arXiv preprint arXiv:2110.08845*.

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

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*, 9:211–225.

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

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*, pages 1515–1520.

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

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)*, pages 514–522.

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*, pages 7871–7880.

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)*, pages 684–695.

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*, pages 1685–1697.

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

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*, pages 4110–4119.

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)*.

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*, pages 2505–2512.

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*, pages 1949–1954.

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)*, volume 1, pages 1073–1083.

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*, pages 4187–4195.

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*.

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*, pages 1784–1795.

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*, pages 1342–1351.

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*, pages 1865–1874.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *arXiv preprint arXiv:2106.11520*.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Wingki Leung. 2020a. Aser: A large-scale eventuality knowledge graph. In *The Web Conference 2020-Proceedings of the World Wide Web Conference, WWW 2020*, page 201.

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

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020c. 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*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

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*.

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*, pages 6236–6247.

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*, pages 1049–1058.

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*, pages 5905–5921.