ClusterVote: Automatic Summarization Dataset Construction with Document Clusters

Daniil Chernyshev(✉) and Boris Dobrov

Research Computing Center, Lomonosov Moscow State University, Moscow, Russia
chdanorbis@yandex.ru

Abstract. Creating a summarization dataset is a costly task due to the amount of expertise and human work required to compose quality summaries. To alleviate the issue, several pseudo-summary approaches were developed, but due to a lack of domain adaptation mechanism, they were not applied beyond language model pretraining. We find that this shortcoming can be overcome by leveraging document clusters. We propose ClusterVote, a pseudo-summarization approach that accounts for domain summarization patterns by studying links between related documents. The method can be configured for different levels of granularity and produce both extractive and abstractive summaries. We evaluate the approach by collecting Telegram news summarization dataset and testing state-of-the-art models. The experimental results show that the most refined variant of ClusterVote has similar extractive properties to CNN/Daily Mail dataset and proves to be challenging for summarization systems.

Keywords: Abstractive summarization · Dataset for summarization · Clustering

1 Introduction

Text summarization is one of the most challenging and demanded domains of Natural Language Processing. The task can be conceptualized as text compression which has two approaches: extractive and abstractive. Extractive summarization leverages existing text fragments to produce a compression that would retain all necessary information. Abstractive summarization expands the extractive approach with additional language resources to paraphrase the extractive fragments into the most concise summary. Training an abstractive summarization system requires a collection of examples with quality summaries. The most common approach to obtaining such data is hiring human experts. However, the amount of work and expertise required to compose quality summaries impose high costs on the procedure. A cheaper alternative would be leveraging resources with prewritten summaries, but such solution is feasible only for a limited selection of domains. Furthermore, significant stylistic and layout differences between subdomains exacerbate

S. R. M. Prasanna et al. (Eds.): SPECOM 2022, LNAI 13721, pp. 99–113, 2022.
https://doi.org/10.1007/978-3-031-20980-2_10

the situation due to strong pattern bias which later dominates the training signal of the summarization systems [9], making them unsuitable for transfer learning.

To alleviate the issue, several universal automatic dataset construction procedures have been proposed [19–21,23]. The idea is to exploit sentence-wise statistics to determine extraction patterns and use the selected sentence subset as a pseudo-summary. This approach proved to be efficient for pretraining language models for abstractive summarization that produce current state-of-the-art results in both supervised and unsupervised settings, such as Pegasus [21] and PRIMERA [19]. However, the heuristics used in existing pseudo-summary methods are human-agnostic and, thus, the extracted sentence subset may significantly differ from the real summary.

Given the source document only, we are limited to the author's viewpoint and cannot derive an unbiased pseudo-summary that would align with the average reader's perception. However, a global salience of source content can be approximated by studying connections with related documents. By selecting the most cited/mentioned sentences we can obtain an objective extractive pattern that would reflect community interest. Following that idea, we propose a new method for pseudo-summary construction - ClusterVote. Unlike previous approaches, our method can produce both extractive and abstractive summaries of variable granularity and accounts for domain on the community level. Using this method, we build Telegram News dataset for abstractive summarization based on data for Telegram Data Clustering Contest 2020[1]. We evaluate different variations of the method by comparing them with the previously proposed pseudo-summary baseline [2] in terms of task complexity and factuality. The results show that the most refined version of ClusterVote has similar extractive properties to popular CNN/Daily Mail dataset [13] and poses the most challenge to state-of-the-art models.

2 Related Work

2.1 News Summarization Datasets

Datasets were always the cornerstone of abstractive summarization research. Historically the first large-scale dataset for mixed summarization was The New York Times Annotated Corpus [18] with several hundred thousand articles written between 1987–2007 that have paired summaries composed by library scientists. However, due to low accessibility and lack of attention in the scientific community that dataset was overshadowed by CNN/Daily Mail [13]. Originally introduced for question answering by Herman et al. [7], CNN/Daily Mail was adapted for abstractive summarization by Nallapati et al. [13] and since then it served as the standard for abstractive summarization evaluation. Due to low abstractiveness of this dataset Narayan et al. [14] later proposed Xsum dataset with extremely compressed summaries for BBC articles. At the same time, Grusky et al. [6] addressed the lack of publisher diversity by scraping HTML metatags from web articles of 38 publishers and constructing Newsroom dataset.

[1] https://contest.com/docs/data_clustering2.

2.2 Pseudo-summary Methods

Summarization datasets contain layout and stylistic biases that differ between subdomains [9]. This makes knowledge transfer inefficient and implies that the dataset should be constructed exclusively for the domain. But due to high costs of manual obtainment researchers sought to develop a universal automatic solution. Several pseudo-summary dataset construction methods were proposed but they were generally employed in pretraining. One of the first to apply the idea to train the language model for summarization was Yang et al. [20]. In their TED model, they exploited the inverted pyramid[2] concept and used the leading sentences of the article as a summary. Zhang et al. [21] showed that this strategy performed worse than random sampling and proposed Pegasus model that utilizes ROUGE [10] score to find the most representative sentences as a proxy summary for summarization pretraining. Xiao et al. [19] followed pyramid evaluation method [15] to expand Pegasus approach to multi-document summarization and use it to pretrain PRIMERA summarization model. Zhong et al. [23] introduced event-based summarization pretraining that aims to recover randomly masked sentences given their event descriptions as a prefix for the input text and, thus, train the language model for controlled summary generation.

3 Constructing Dataset with ClusterVote

As the basis for our experiments, we chose data provided for Data Clustering Contest 2020 hosted by Telegram. The goal of the contest was to cluster news articles based on various features: language, categories, and topics. The result of this process are news threads which essentially should contain only contextually similar documents. Based on the contextual similarity assumption, we develop a ClusterVote method that ranks sentences based on popularity of presented information.

3.1 Telegram Data Clustering Contest 2020 Dataset

There are multiple languages in Telegram Data Clustering Contest dataset, however, we processed only the English part. This part has more than 560 000 articles from 1346 publishers covering a time span of 52 days. Since Telegram did not provide ground truth labels for the contest data, the clusters were collected manually[3] To ensure coherence, we filtered out all data that had less than 50 words in the main body and had more than 30% of numeric characters. Afterward, we followed a simple iterative clustering procedure.

To optimize clustering performance, all data was split into 72-hour buckets with 24-hour overlap to take into account the possible 48-hour lag that is typical for analytical articles. Each bucket was clusterized in two steps using DBSCAN [4] algorithm with cosine distance. First, all articles in the bucket were clusterized

[2] https://en.wikipedia.org/wiki/Inverted_pyramid_(journalism).

[3] All scripts used for dataset collection are available at: https://github.com/dciresearch/ClusterVote.

according to mentioned named entities to isolate different subjects. Then, these clusters were broken down into smaller event-wise subclusters with standard tf-idf vectors on leading 4 sentences of the article. At this stage recall is more important than precision as our method will further refine the clusters.

3.2 ClusterVote Method

The document context can be described as a set of textual facts. This means that contextual similarity can be interpreted as similarity of presented fact sets. Following that logic, the document clusters are formed around some factual core that remains constant regardless of document. The distance of individual fact to this core reflects how interested is the community. The closest (zero-distance) facts were ranked as salient by majority of authors and can be considered as the basis for objective document-wise summary, while the farthest facts were omitted in most documents likely due to their supplementary nature.

Given that facts are naturally grouped into sentences, the cluster's factual core can be represented as the most common similar sentence subset. This subset can be identified by pairing each sentence with sentences of all other documents in the cluster and then calculating the relative number of support the sentence received:

$$vote(s_i) = |\{D_k \mid s_i \notin D_k, \exists s_j \in D_k : s_j \equiv s_i\}| \quad (1)$$

where D_k is k-th document of cluster and s_i is sentence. By selecting sentences that received support over some threshold we can obtain a cluster summary from document's viewpoint. Varying the sentence selection threshold would yield summaries of different levels of granularity. Naturally, sampling sentences over the same document would yield an extractive summary. To obtain an abstractive summary we can sample the sentences from any other document in the cluster. As a side effect, this process identifies cluster outliers and identical articles, dropping which will improve overall clustering quality and will ensure contextual similarity. In some sense, the documents are voting for sentences from other documents in the cluster, hence the name of our method, ClusterVote.

To pair sentences, we used DBSCAN algorithm with cosine distance and sentence embeddings obtained by paraphrase-mpnet-base-v2 model from Sentence-Transformers [17]. To avoid redundancy, connections within the same document were blocked by setting distances to ∞. Since both total number and diversity of cluster sentences can vary, we dynamically set the neighborhood threshold ε with grid search to guarantee that the maximum distance within the cluster would not exceed the paraphrase threshold $d_{par} = 0.2$. We classified as outliers all articles in which sentences had been paired only with articles that had more than 60% of zero vote sentences and as identical which had more than 80% of similarly paired sentences. We produced two variants of summaries:

- CV-full – vote threshold $t_{vote} = 1$
- CV-max – vote threshold $t_{vote} = \max(\{vote(s_i) \mid s_i \in \text{Article}\})$

In each clustering pair the article to serve as a source of summary sentences was chosen according to the ratio of paired sentences.

Comparison with LexRank. The ClusterVote method can be thought of as a specific variant of unweighted LexRank [3] with additional edge filtering. Such filtering can be conceptualized as a reduction of a sentence similarity graph to a cluster graph where the only remaining connected components are cliques. Since each node's degree centrality scores cannot be influenced by nodes other than those in clique, LexRank is guaranteed to preserve clique-wise centrality and, thus, produce the same ranking as ClusterVote.

3.3 Dataset Statistics

After all clustering and filtering procedures, we were left with 110 713 source-summary pairs. For convenience, we name the resulting dataset Telegram News. We split our summarization dataset into training (80%, 88 573), validation (10%, 11 070), and test (10%, 11 070) sets. Since our dataset uses artificial summaries, their human-likeliness should be evaluated. The simplest way to address that is to compare to existing human collected datasets. Table 2 compares our dataset to popular summarization datasets. We chose CNN/Daily Mail since it is the standard for testing single document summarization systems. Additionally, we compare with Newsroom as its automatic HTML metatag scraping approach is a direct alternative for our ClusterVote method in the internet news domain. Besides ClusterVote summaries, we compare a simple pseudo-summary baseline obtained by taking leading 3 sentences from a paired summary article (denoted as Pseudo Lead). In previous work, this approach proved to be eligible for training abstractive summarization models [2]. Example of pseudo-summaries is provided in Table 1.

In addition to length statistics, we report extractive metrics [6] to give some insights into the abstractiveness of automatic cluster-based summaries. Extractive fragment coverage is a percentage of words in the summary that were retained during the summarization process:

$$\text{Coverage}(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} |f| \tag{2}$$

where A is an article text, S is a summary text, $F(A, S)$ is a set of summary fragments that were extracted from the article and $|t|$ is number of words in text t. This metric demonstrates vocabulary similarity and, thus, word-substitution paraphrases. Extractive fragment density is the average length of extractive fragment f to which each word in the summary belongs:

$$\text{Density}(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} \sum_{w \in f} w \cdot |f| = \frac{1}{|S|} \sum_{f \in F(A,S)} |f|^2 \tag{3}$$

while density looks similar to coverage it is sensitive to the number of extractive fragments. For instance, if we have a summary of length of 60 and the total length of extractive fragments of 42, we will have a density of 29.4 in case of only one extractive fragment and of 14.7 if there are two equally sized fragments while

Table 1. Example of pseudo-summary strategies. ClusterVote variants provide more information than Pseudo Lead, but CV-max completely filters out the citations.

Source URL: https://www.vanguardngr.com/2020/04/ronaldinho-says-arrest-and-confinement-has-been-hard/
Pseudo Lead: Kindly Share This Story: Former Brazilian football star Ronaldinho said on Monday that his arrest and subsequent house arrest in Paraguay for using a false passport was "a very hard blow". "I would never have imagined myself in such a situation," said the former Barcelona and Paris Saint-Germain striker in an interview with the Paraguayan newspaper ABC
CV-full: Former Brazilian football star Ronaldinho said on Monday that his arrest and sub-sequent house arrest in Paraguay for using a false passport was "a very hard blow". Ronaldinho and his brother Roberto de Assis Moreira are accused of entering Paraguay in possession of false passports. "I was completely caught off guard when I found out that these passports were not valid," Ronaldinho said in his first public statement since his arrest two days after arriving in Asuncion at the beginning of March. Since April 7 they have been under house arrest at the Palmaroga Hotel in the historic centre of the Paraguayan capital. Ronaldinho said he hopes to be released "as soon as possible" after cooperating with the Paraguayan police investigation. Ronaldinho and his brother face up to five years in prison if found guilty
CV-max: Ronaldinho and his brother Roberto de Assis Moreira are accused of entering Paraguay in possession of false passports. Since April 7 they have been under house arrest at the Palmaroga Hotel in the historic centre of the Paraguayan capital. Ronaldinho and his brother face up to five years in prison if found guilty

coverage will be 0.7 regardless of fragment partition. The maximum value of extractive density is equal to the length of the summary, thus, to negate dataset summary length distribution effect it should be normalized. The normalized density indicates the proportion of whole sentence paraphrases.

Table 2. Comparison of different automatic summary construction approaches to popular abstractive summarization datasets. "norm." is length normalized values.

Dataset	Source # words	Summary # words	Coverage	Density	
				raw	norm.
CNN/DM [13]	781	56	89%	3.87	0.07
Newsroom [6]	659	27	83%	9.51	0.36
Telegram news					
Pseudo Lead	438	88	87%	26.10	0.29
CV-full		237	91%	43.43	0.18
CV-max		95	92%	28.81	0.30

The first noticeable trait is that our dataset has generally shorter source articles, while summaries are at least two times longer on average than summaries of CNN/Daily Mail and Newsroom. Nevertheless, Newsroom has the highest length normalized average density while CNN/Daily Mail has the lowest. Among our summaries, CV-full has the lowest density but it is more than two times larger than of CNN/Daily Mail and CV-max has the same density as Pseudo Lead. This indicates that lower vote sentences are likely to contain more full-phrase paraphrases. On the other hand, the average coverage is around 90% for all summaries but Newsroom which has 83%. This implies that most paraphrasing in summaries is based on word reordering.

The next question is how faithful is paraphrasing? Factual mistakes are common in abstractive summarization models, and it has been shown that training dataset plays a major role in erroneous text hallucinations [11]. Since summary is a text compression it must contain no other information than the source article. Therefore, factual correctness must have a higher priority than abstractiveness. Currently, there is no universal measure for factuality since fact definition heavily relies on the extracted context. To address different factual aspects, we measure multiple summary-source precision metrics. Phrasal Accuracy complements extractive fragment coverage reflecting a phrase extraction ratio:

$$phrase_{acc}(A, S) = \text{ROUGE-2}_{prec}(S, A) \tag{4}$$

High phrasal extractiveness guarantees that summary sentences align with text's content, while low values indicate excessive paraphrasing or unrelated external information. Named entity overlap (NEO) is a percentage of named entities that have been correctly reproduced in summary:

$$NEO(A, S) = \frac{|NE(A) \cap NE(S)|}{|NE(S)|} \tag{5}$$

where $NE(T)$ is named entities of text T. Named entities define the contextual core and, thus, do not tolerate any distortions in most cases. Factual accuracy is a percentage of summary subject-relation-object fact triplets supported by the source article:

$$fact_{acc}(A, S) = \frac{|Facts(A) \cap Facts(S)|}{|Facts(S)|} \tag{6}$$

$fact_{acc}$ is the basic way to measure factuality in summarization [5]. We use Spacy[4] to extract named entities and OpenIE[5] to extract fact triples for metrics.

The common problem with mentioned metrics is that they are not robust to synonymous substitutions. This renders them misleading in extreme paraphrasing scenarios. To accommodate this case, we report BERTScore [22] summary-article precision which has been shown to have a better correlation with human judgement in FRANK factuality metric benchmark[6] [16].

[4] https://spacy.io/.

[5] https://nlp.stanford.edu/software/openie.html.

[6] https://frank-benchmark.herokuapp.com/.

Table 3 compares datasets in terms of factuality. First of all, our automatic summaries have a significantly higher percentage of extracted phrases than summaries from popular datasets. This means that our summaries are less likely to contain unsupported content. But, despite higher extractiveness, Pseudo Lead is more erroneous in terms of named entity reproduction than more abstractive counterparts, being the most inaccurate in the comparison set. Since NEO and $fact_{acc}$ require exact matches, it was expected that the most extractive approach, CV-max, demonstrates the highest results. However, according to $fact_{acc}$ CNN/Daily Mail has the most unfaithful summaries. This is the result of metric strictness that penalizes any paraphrasing. Since CNN/Daily Mail has the lowest extractiveness metrics, with a normalized average density of 0.07, n-gram-based measures are ineffective. Therefore, auxiliary embedding-based metrics are essential. According to BERTScore CNN/Daily Mail still has the lowest factuality, though it is marginally lower than Newsroom. Unexpectedly, CV-full method exhibits the highest BERTScore factuality almost twice of CNN/Daily Mail and 6% more than Pseudo Lead and CV-max. Considering lower extractiveness and $fact_{acc}$ values, this result reaffirms the hypothesis that lower voted sentences are likely to be more paraphrased.

Table 3. Factuality statistics for datasets.

Dataset	$phrase_{acc}$	NEO	$fact_{acc}$	BERTScore
CNN/DM [13]	49.78%	78.12%	9.39%	36.28%
Newsroom [6]	53.89%	76.42%	38.39%	38.42%
Telegram news				
Pseudo Lead	72.10%	75.53%	44.57%	63.41%
CV-full	79.28%	80.18%	54.57%	**69.77**%
CV-max	**83.09%**	**84.38%**	**61.90%**	63.15%

4 Evaluation

The main concern about any dataset is "how eligible it is for the task?" or "how trivial is the solution pattern?" Answering that question requires experimenting with systems of various complexity. A good dataset should avoid two extremes: a low performance of the most advanced approaches would indicate inconsistency in solution patterns that is likely to be attributed to noisy examples [8], while high performance of the simplest strategies will reveal strong pattern bias [14]. Additionally, since our summaries were obtained automatically by leveraging external information (cluster of related documents), by benchmarking solutions we are also studying how efficiently this information can be derived from the source document only. We evaluate the performance of state-of-the-art abstractive summarization models in both supervised and unsupervised settings and compare them to extractive baselines.

4.1 Extractive Baselines

Lead-k. The most common baseline in text summarization is taking k leading sentences of the source document as a summary. Usually, those sentences introduce facts essential for document context comprehension. However, in the news domain leading sentences can cover the whole summary. This is due to the widespread inverted pyramid news writing scheme, that ensures that the information is presented in the order of salience. According to the scheme, the first paragraph must contain the minimum information set to give the reader idea of the story. The first paragraph usually consists of 2–4 sentences, hence the common value $k = 3$. This strategy is considered to be a lower performance bound for news summarization.

Oracle. The popular method for approximating an upper performance bound for text summarization is the greedy oracle. The idea is to iteratively sample source sentences to maximize the reference summary similarity metric.

TextRank. TextRank is a sentence-level extractive summarization system proposed by Mihalcea et al. [12]. TextRank was proposed at the same time as LexRank and is based on the same PageRank [1] algorithm but was designed for single document summarization. TextRank defines the lower performance bound for extractive summarization systems.

Table 4. Model performance comparison for Pseudo Lead summaries.

Model	ROUGE-1	ROUGE-2	ROUGE-L	# words
Extractive baselines				
Lead-3	74.41%	64.31%	68.57%	84
Oracle	80.16%	70.67%	74.40%	85
TextRank	47.46%	30.89%	37.81%	92
Unsupervised				
PRIMERA	36.49%	19.33%	25.63%	146
Pegasus	42.68%	28.23%	34.62%	79
Supervised				
PRIMERA	72.13%	62.66%	66.43%	102
Pegasus	76.17%	66.53%	70.68%	90

4.2 Abstractive Summarization Models

At the moment of writing this article, Pegasus [21] and PRIMERA [19] models achieve state-of-the-art results in unsupervised abstractive summarization. Both models were specifically pretrained for abstractive summarization, but

PRIMERA specializes in multi-document setting and has four times larger input limit thanks to sparse attention. In a supervised setting both models achieve the same or comparable results to state-of-the-art methods. Fine-tuning these models will provide insights into similarity of automatic summarization dataset construction approaches and the effects of pretraining biases.

4.3 Setup

In the Oracle baseline we maximize the average of ROUGE-1 and ROUGE-2 F1 scores. In Lead-k we choose $k = 3$ for Pseudo Lead and CV-max and $k = 6$ for CV-full. TextRank number of extracted sentences is controlled by desired summary length which we configured to the average length of reference summaries. For a fair comparison of abstractive summarization models, we limit input text to Pegasus maximum input length of 1024 tokens. To generate summaries, we use beam search with beam size 5 and trigram blocking. We report ROUGE F1 scores and prediction-source precision factuality metrics: $phrase_{acc}$, NEO, $fact_{acc}$, and BERTScore.

4.4 Summarization Metrics

Table 4 reports Pseudo Lead reference summaries results on test set. As expected, Lead-3 baseline performs better than more complex non-Oracle methods. However, only 64% of phrases come from article lead and the best extractive strategy won't cover more than 70%. Due to strong extractive positional bias, universal unsupervised methods will significantly underperform. Since PRIMERA was trained for multi-document summarization it is biased towards longer summaries and produces more irrelevant content than Pegasus in unsupervised setting. Even though fine-tuning improves the performance, the length bias is still strong, preventing the PRIMERA from surpassing the Lead-3 baseline.

Table 5. Model performance comparison for CV-full summaries.

Model	ROUGE-1	ROUGE-2	ROUGE-L	# words
Extractive baselines				
Lead-6	56.96%	47.49%	50.24%	176
Oracle	86.15%	78.70%	80.87%	227
TextRank	58.28%	44.60%	47.77%	218
Unsupervised				
PRIMERA	45.60%	30.25%	32.81%	146
Pegasus	38.00%	28.01%	31.95%	79
Supervised				
PRIMERA	66.91%	57.24%	60.02%	162
Pegasus	65.47%	56.29%	58.80%	160

The results for CV-full evaluation are presented in Table 5. Despite longer summaries, the upper bound for extractive method performance is significantly higher with Oracle average ROUGE-2 score of 79%. Lead baseline is noticeably less efficient in this case which indicates a more uniform salient sentence distribution. PRIMERA outperforms Pegasus in both supervised and unsupervised settings, however, the difference for fine-tuned models is marginal. This suggests that lower PRIMERA performance on Pseudo Lead summaries was attributed to the model's length bias.

CV-max performance is reported in Table 6. Despite similar average reference summary length, Lead-3 baseline has the worst performance in this case, covering only 39% of summary phrases. In contrast, Oracle achieves the maximum extractive phrase coverage of 81%. These two facts combined mean that CV-max summaries are indeed extractive but require complex strategies for sentence sampling. For PRIMERA this type of reference is slightly more familiar than Pseudo Lead, meanwhile Pegasus summaries differ the most. However, just like with Pseudo Lead, Pegasus outperforms PRIMERA in both supervised and unsupervised settings due to the same length bias persistence of the latter. Overall performance of fine-tuned abstractive summarization model in CV-max scenario is similar to CV-full which indicates the consistency of extractive patterns during vote threshold reduction.

Table 6. Model performance comparison for CV-max summaries.

Model	ROUGE-1	ROUGE-2	ROUGE-L	# words
Extractive baselines				
Lead-3	52.07%	38.60%	43.56%	84
Oracle	87.00%	81.14%	83.72%	85
TextRank	43.04%	26.73%	33.70%	92
Unsupervised				
PRIMERA	37.03%	21.60%	27.32%	146
Pegasus	39.12%	23.71%	30.87%	79
Supervised				
PRIMERA	60.92%	51.36%	54.87%	114
Pegasus	64.91%	55.33%	59.09%	88

To study the extractive patterns, we report Oracle extracted sentence relative distribution in Fig. 1. As it can be seen, Pseudo Lead is extremely skewed towards the first sentences of the article which explains extraordinary Lead-3 performance. As was noted earlier, CV-full is significantly more balanced, showing almost uniform distribution for the first half of the source sentences, however, the probability of sampling sentences after that point falls sharply. CV-max summaries are the most natural of all, bearing the most extractive pattern similarity to CNN/Daily Mail. Considering that CV-max is a more refined version of CV-full, we can conclude that lower voted sentences are sentences that were frequently filtered out from the original material or rarely placed at the top of inverted pyramid. The inverted pyramid scheme also explains the undersampling of concluding sentences as those are used to provide optional comments or background information about subjects and events. This does not imply that this information cannot be salient, as CNN/Daily Mail distribution demonstrates the otherwise, however, the relevance of this content is strictly dependent on the reader's knowledge.

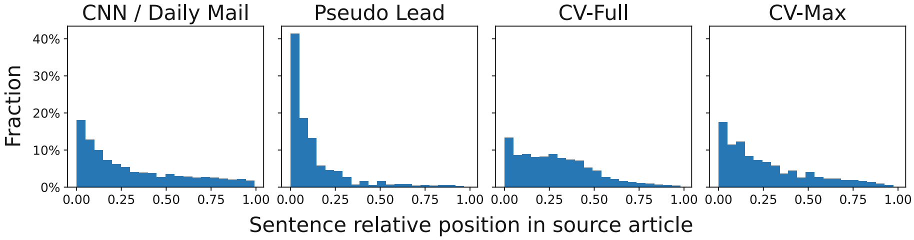

Fig. 1. Oracle extracted sentence relative position distribution for each summary type. We use CNN/Daily Mail as the baseline for comparison since most previous work on dataset biases studied the effect of that dataset on summarization model performance.

4.5 Factuality Metrics

To assess dataset factuality bias, we compare factuality metrics before and after finetuning. If examples in the training set promote extrinsic hallucinations (not directly deducible from the source text), we will observe token-wise factuality degradation for models that were specifically pretrained to leverage source text facts only. Pegasus was trained to recover sentences that had the most common extractive fragments with a text remainder. Given that these extractive fragments contain the same phrasal form of facts and entities, Pegasus is expected to have high NEO and $fact_{acc}$ values. Since PRIMERA expands Pegasus approach for multi-document clusters and prioritizes named entity frequency over extractive density and does not employ any source-summary fact connection verification procedures, the model is likely to be pretrained on both more abstractive and noisier summaries and, therefore, produce less faithful texts.

Table 7. Factuality scores for abstractive summarization models.

Model	$phrase_{acc}$	NEO	$fact_{acc}$	BERTScore
Unsupervised				
PRIMERA	86.74%	90.69%	76.47%	54.65%
Pegasus	98.13%	98.74%	93.42%	70.16%
Supervised - Pseudo Lead				
PRIMERA	87.97%	85.74%	64.83%	73.10%
Pegasus	93.74%	90.12%	74.82%	74.62%
Supervised - CV-full				
PRIMERA	91.61%	88.64%	69.76%	79.78%
Pegasus	95.99%	93.26%	80.35%	84.58%
Supervised - CV-max				
PRIMERA	96.02%	95.04%	82.51%	75.95%
Pegasus	96.44%	95.49%	83.10%	79.71%

Table 7 provides the results of our factuality measurements. Just as hypothesized, PRIMERA has a lower factuality than Pegasus. However, its lower NEO was not expected as the model was trained specifically to consider named entities in cases where Pegasus would underestimate their importance. Unsupervised Pegasus has an almost 100% named entity reproduction rate and over 93% fact triplet overlap which is explained by 98% phrasal overlap. However, BERTScore at 70% suggests that summaries have altered context meaning likely due to phrase permutation or omitted words.

Fine-tuning models on our datasets shifts the extractive behavior. In all cases, PRIMERA learns to follow the text more carefully, while Pegasus starts to paraphrase. Despite the positional bias, the least extractive summaries are produced after fine-tuning on Pseudo Lead summaries. On the other hand, with this type of summary we observe the strongest factuality decline for both models. Considering that CV-max reference summaries promote the most extractive behavior and yet have higher BERTScore, Pseudo Lead summaries are likely to be inconsistent with the source article. CV-full holds the middle ground, having the best BERTScore and more abstractive models than CV-max which again confirms "lower vote - more abstractive" hypothesis. Interestingly, all generated summaries have substantially higher factuality and extractiveness than references they were learning to replicate. This hints an existence of strong extractive bias in pretrained summarization models that prevents them from learning abstractive patterns.

5 Conclusion

In this work, we proposed ClusterVote, a new method for automatic construction of a summarization dataset that, in contrast to previous approaches, produces pseudo-summaries with objective extractive patterns by leveraging document cluster connections. Using the method, we construct a Telegram News dataset. We evaluated two extreme cases of ClusterVote granularity: CV-full – sentences with at least one vote, and CV-max - the most refined version with maximum cluster support. We compared these pseudo-summaries to our previous Pseudo Lead baseline that exploits inverted pyramid news structure and alternative publishers. According to statistics, CV-max is the most extractive of all yet has the closest resemblance to human-annotated datasets like CNN/Daily Mail in terms of sentence extraction pattern. CV-max summaries also show the highest factuality, however, CV-full lower measurements are likely attributed to higher abstractiveness which was consistently suggested during our experiments. Despite that, both strategies were found fairly difficult for state-of-the-art abstractive summarization models, achieving only 66% ROUGE-1 after fine-tuning. The equivalence of extreme cases suggests that max-full interpolation will be also eligible for summary proxy. The models themselves have demonstrated a biased behavior during the fine-tuning process with PRIMERA failing to abolish summary length bias and Pegasus hardly deviating from familiar from pre-training extractive strategy. We believe that both bias and low abstractiveness issues can be overcome by data scaling as larger clusters will have more diverse voting patterns as well as better paraphrase ranking.

Acknowledgments. The work of Daniil Chernyshev (experiments, survey) was supported by Non-commercial Foundation for Support of Science and Education "INTELLECT". The work of Boris Dobrov (general concept, interpretation of results) was supported by the Russian Science Foundation (project 21-71-30003).

References

1. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**, 107–117 (1998)
2. Chernyshev, D., Dobrov, B.: Abstractive summarization of Russian news learning on quality media. In: van der Aalst, W.M.P., et al. (eds.) AIST 2020. LNCS, vol. 12602, pp. 96–104. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72610-2_7
3. Erkan, G., Radev, D.R.: LexRank: graph-based lexical centrality as salience in text summarization. J. Artif. Int. Res. **22**(1), 457–479 (2004)
4. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (1996)
5. Goodrich, B., Rao, V., Liu, P.J., Saleh, M.: Assessing the factual accuracy of generated text. In: proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 166–175 (2019)
6. Grusky, M., Naaman, M., Artzi, Y.: Newsroom: a dataset of 1.3 million summaries with diverse extractive strategies. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 708–719 (2018)
7. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems 28 (2015)

8. Kang, D., Hashimoto, T.B.: Improved natural language generation via loss truncation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 718–731. Association for Computational Linguistics (2020)
9. Kryściński, W., Rajani, N., Agarwal, D., Xiong, C., Radev, D.: BookSum: a collection of datasets for long-form narrative summarization. arXiv:2105.08209 (2021)
10. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
11. Maynez, J., Narayan, S., Bohnet, B., McDonald, R.: On faithfulness and factuality in abstractive summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1906–1919 (2020)
12. Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (2004)
13. Nallapati, R., et al.: Abstractive text summarization using sequence-to-sequence RNNs and beyond. In: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp. 280–290 (2016)
14. Narayan, S., Cohen, S.B., Lapata, M.: Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1797–1807 (2018)
15. Nenkova, A., Passonneau, R.: Evaluating content selection in summarization: the pyramid method. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, pp. 145–152 (2004)
16. Pagnoni, A., Balachandran, V., Tsvetkov, Y.: Understanding factuality in abstractive summarization with FRANK: a benchmark for factuality metrics. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4812–4829. Association for Computational Linguistics (2021)
17. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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), pp. 3982–3992 (2019)
18. Sandhaus, E.: The New York times annotated corpus (2008)
19. Xiao, W., Beltagy, I., Carenini, G., Cohan, A.: PRIMERA: pyramid-based masked sentence pre-training for multi-document summarization. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 5245–5263 (2022)
20. Yang, Z., et al.: TED: a pretrained unsupervised summarization model with theme modeling and denoising. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1865–1874 (2020)
21. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning, pp. 11328–11339. PMLR (2020)
22. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: evaluating text generation with BERT. arXiv preprint arXiv:1904.09675 (2019)
23. Zhong, M., et al.: Unsupervised summarization with customized granularities. arXiv preprint arXiv:2201.12502 (2022)