Sustainable Computing: Informatics and Systems 32 (2021) 100607

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Sustainable text summarization over mobile devices: An energy-aware approach

Krishnandu Hazra [a,*], Tanmoy Ghosh [a], Avirup Mukherjee [b], Sujoy Saha [a], Subrata Nandi [a], Saptarshi Ghosh [b], Sandip Chakraborty [b]

[a] National Institute Of Technology Durgapur, India
[b] Indian Institute Of Technology Kharagpur, India

ARTICLE INFO

Keywords:
On-device summarization
Module-wise energy estimation
Energy-aware text summarization
Sustainable computing
Energy constraint scenarios

ABSTRACT

Due to the omnipresence of hand-held mobile devices, people nowadays are running many applications in such devices to fulfill their daily life requirements. However, due to the limited energy (battery power) of mobile hand-held devices, the energy consumption of such applications determines their feasibility of running in such mobile devices for a long term basis. One such important application is the summarization of text information. Although almost all the existing summarization approaches to date are designed to run on high-end servers or cloud platforms aiming to optimize only the summary quality, there are many applications nowadays, e.g., summarizing data in crisis scenarios or summarizing privacy-sensitive data which demands the functionality of on-device *computationally light-weight* text summarization to generate effective summaries, while keeping in mind the limited battery power of the device. This paper is the first of its kind where we design energy-efficient summarization algorithms for mobile devices. First, we provide a methodology to systematically measure the energy consumption characteristics of various classical extractive summarization techniques at a modular level by analyzing the algorithmic constructs. Through this process, energy-hungry modules are identified under different configurations and environmental parameters, like input data type, dataset size, device type, among others. Next, based on the observations, we develop four classes of energy-efficient hybrid summarization approaches. Extensive experiments show that the hybrid summarization approaches, when applied on various datasets of varying size and type, can save up to 90% energy, with 5–40% degradation in the summary quality with respect to the high-performing base summarization approaches.

1. Introduction

Smartphones and various other hand-held devices are increasingly being used to create and access content over the Web [27]. Given the information overload on the Web today, often a user may be only interested to look into a summary of the contents, e.g., the highlights of the today's news articles based on her personalized choices, a summary of messages she received over social media or on her messaging application, a spotlight of the emails she received during the last night, and so on. Several smartphone applications, such as *Inshorts*,[1] *Dailyhunt*,[2] and SMS or messenger apps with personalized content highlighting already provide such summarization functionality.

In this context, there exist a large number of summarization algorithms [14,21,24] that can be used to summarize text. Almost all the existing summarization algorithms assume that the algorithm will be executed on server/cloud platforms. As a result, the smartphone applications like *Inshorts*, *Dailyhunt* which offer summarization functionality, also connect to a central server or cloud platform to execute the summarization procedure and provide the summary to the device. Due to the abundance of energy and computation power of the servers, the summarization algorithms solely attempt to optimize the quality of the summary.

However, a text summarization algorithm may also need to be executed on hand-held devices. Several practical scenarios motivate such on-device summarization, some of which are as follows: To ensure privacy and confidentiality, a user may prefer not to directly upload

* Corresponding author.
E-mail address: krishnandu91@gmail.com (K. Hazra).

[1] https://inshorts.com/.
[2] http://www.dailyhunt.in/.

https://doi.org/10.1016/j.suscom.2021.100607
Received 10 August 2019; Received in revised form 8 August 2021; Accepted 29 August 2021
Available online 3 September 2021
2210-5379/© 2021 Elsevier Inc. All rights reserved.

content from his/her mobile device to a cloud platform, rather summarize the content on his/her device and then upload only the summary. Again, during the disaster scenario (e.g., a hurricane/earthquake) where infrastructurally it might be hard to provide electrical power and network connectivity (Cellular/Internet) in the long term, there have been attempts to form a temporary ad-hoc communication network over the hand-held devices ([15], Teamphone [39], NICER911 [44]) equipped with data collection capability for collecting the information in the affected region for rescue/relief need assessment and disaster recovery. In such a setting, each mobile device is expected to gather information from nearby devices and communicate with each other seamlessly using Delay Tolerant Network (DTN) opportunistic networks. However, it has been observed that there is a flood of information during the large-scale disaster because a ton of information occurs in a short time [2]. Hence, in such a scenario, to exploit such a large amount of data to extract meaningful information, frequent on-device summarization is essential. This on-device summarization will help eliminate redundant data communication and use the communication bandwidth to the fullest extent by transmitting only the non-redundant information.

However, a significant limitation of such hand-held devices is limited energy (battery power). Thus, to run in such hand-held devices, the text summarization algorithms need to be energy-efficient. Moreover, in addition to being energy efficient, the summarization algorithms must maintain an acceptable summary quality to generate relevant summaries. Therefore, a trade-off between energy consumption and summarization performance[3] must be maintained to create a sustainable summarization algorithm for hand-held devices. This is non-trivial because the computation vs. limited energy (battery power) trade-off in hand-held devices has always been a paramount challenge [54]. Also, reducing the power consumption of computation is an important research area today, termed green computing. More specifically, our work is aligned to the increasingly popular research on green computing for mobile devices [3,29], which is needed in today's world where there is an omnipresence of hand-held devices to process data.

A few works have attempted to design on-device summarization approaches [5,6,9,10,36]; but, most of them [5,6,9] have utilized the hand-held devices only to visualize the summary, while the summarization algorithm runs on some remote servers. Some works [9,10] have proposed on-device summarization and classification of tweets. However, none of the existing works considered the challenges of designing on-device summarization.

To the best of our knowledge, sustainable text summarization algorithms suitable for such hand-held devices have not been studied systematically yet. To this end, we ask the following questions in this work:

1. Can the existing summarization algorithms be executed on mobile hand-held devices sustainably? Considering that many summarization algorithms exist, which one would then be preferred?
2. How can one accurately estimate the energy consumption behaviour of summarization algorithms over mobile devices?
3. Can one develop an energy-efficient summarization approach, that minimizes energy consumption while also computing a good summary?

There are two broad classes of summarization techniques – extractive [24,30] and abstractive [47]. While extractive text summarization approaches extract the most important phrases or sentences from the input text, abstractive summarization methods generate their own summary by rephrasing the input text. Given that abstractive summarization requires complex learning and huge resources for execution [47], they are not suitable for execution over mobile hand-held devices. Hence, in this work, we focus specifically on extractive summarization algorithms.

In this paper, we measure the energy consumption and performance of six classical extractive text summarization approaches [20,40,49,55,59,61] (described in Section 3) over two different Android mobile platforms – one low-end phone and one mid-end phone. We develop a methodology using *Monsoon power monitor* [57] to measure the power consumption of the summarization algorithms at the modular level, by breaking the procedures in four generic steps (Section 4). We perform extensive experiments on two different types of datasets as the input to the summarization approaches – microblog (tweet) datasets related to disaster events, and the DUC 2007 datasets that are popular benchmarks for document summarization.[4] We observe that there is a trade-off between the energy-efficiency and the summary quality for different summarization methodologies (Section 5). We specifically identify those modules that are energy-hungry. Based on these observations, we develop four classes of energy-efficient hybrid summarization approaches. The basic philosophy behind the hybrid summarization approaches is to use an energy-efficient but low-performing algorithm to generate an intermediate summary, and then use a high-performing algorithm (on the intermediate summary) to produce the final summary. From the results (Section 7) we observe that while maintaining a tradeoff between the energy consumption and the summary quality the proposed hybrid summarization approaches always outperform the base summarization approach.

In a nutshell, the contributions of this paper are as follows:

1. To the best of our knowledge, this is the first work that systematically measures the energy consumption characteristics of various summarization algorithms on mobile devices for sustainable computing.
2. We identify the algorithmic modules which are most energy-hungry during summarization. We explore the energy consumption characteristics of those modules under different configuration and environmental parameters, like input data type, data size, device type, summary requirements, among others.
3. Based on the observations from module-level analysis, we define four classes of hybrid summarization approaches which are energy-efficient and produce a good quality summary on mobile devices.

2. Related work

Text summarization: Research on text summarization [19] has been in existence since the mid-20th century starting from Luhn [40]. With the years passing, there have been lots of different approaches like, graph-based [20,41], semantic-based [59], optimization-based [1,55], fuzzy-logic-based [28] among others. Additionally, there are several application-level summarization techniques, like microblog summarization [37,53,55,62]. In recent years, neural network-based summarization algorithms have been developed [18,58]. Also, with the advent of transformer-based language models, there has been researches [38,66] on how pre-trained models like BERT [16] can be used for text summarization. However, fine-tuning such models require high-end GPUs. The prime rationale of all these approaches has been to improve the quality of the summary and not energy-efficiency. Especially, neural summarization models require huge amounts of energy [33], and hence are not suitable for mobile devices; hence we do not consider such neural models in this work.

Text summarization on mobile devices: Most works on summarization over mobile and hand-held devices have only focused on summarizing web contents and effectively visualizing the content in the mobile device [5,6,11]. The works in [50,65] described a hierarchical text summarization and a fractal summarization model that enables users to view Web document summaries in small, portable devices. However, the summarization is performed remotely and sent to the mobile device as per the user's demand, and the mobile device is only

[3] The summarization performance is proportionate with the computational complexity of the summarization algorithms.

[4] https://duc.nist.gov/duc2007/tasks.html.

used for viewing purposes. Similarly, the works [32,52] have explicitly mentioned that mobile devices are incapable of running high computational summarization algorithms, thus have performed the summarization on servers. To the best of our knowledge, none of these works looks into the energy consumption aspects, while executing the summarization procedure over a mobile device.

A few works have attempted to design on-device summarization approaches. The works [5,6] provide very crude summaries by only extracting the most frequent words. The works [7,23] have created mobile applications for summarization, but sentence scoring is based solely on the basic metrics such as Sentence Position, Word Frequency, Sentence Length, Numeric Data, TF/IDF, which provides a rudimentary summary. The work [9,10] does tweet categorization and produces a summary for each category in handheld devices. The work [36] uses a graph-based computationally expensive approach of summarization, and they have not done any energy analysis of their approach. Similarly, in [22], Latent Semantic Analysis (LSA) algorithm has been applied in mobile devices, but no energy consumption analysis is performed. More importantly, the methodologies developed in [5,6,9,10,36] are meant to be computationally efficient (i.e., have a small execution time), but *not* done any energy consumption analysis. We show later in this paper that having a small execution time does *not* necessarily imply that the method is energy-efficient.

Sustainable computation for hand-held devices: Due to the limitation of battery power, handheld devices need energy-efficient applications that sustain their computation for a long time to fulfil user requirements. There are many kinds of research about designing mobile applications in an energy-efficient way [13,42,56] for mobile devices. The authors in [13] described how to implement mobile applications in energy-efficient ways. They provided two kinds of energy-saving techniques such as Method Reallocation (placing the code in different scopes of execution within a single target) and Method Offloading (placement of code in external resources in different scopes) and also identified which method is convenient for processing a particular task in an energy-efficient way. The study [56] revealed a connection between energy consumption and the object-oriented structural metrics of Android applications. Again, the authors in [42] provided a literature review of the existing energy-saving approaches for developing mobile applications. Although, there are many such works (as discussed above) in the literature about designing energy-efficient mobile application in general, however, as most of the existing summarization approaches are server/cloud-based, hence the field of energy-aware summarization for mobile devices is not unveiled by the researchers.

Energy consumption of mobile-based algorithms: Though not for summarization, energy consumption behaviours of various applications and algorithms running on mobile devices have been discussed in the literature. According to the researches [17,26,48], there are several methods for measuring energy consumption in Android OS. However, since the calculation methodology is not standardized, it is difficult to execute and compare them. Most of them are only modelling the energy estimation with respect to some selective parameters rather than measuring energy consumption. Thus, there is still a gap for a readily available tool [48], which will be used for calculating energy consumption by an application in Android OS. In [12], an experimental study disclosed the power usage features of some data mining algorithms running on hand-held devices. This work measures energy consumption from battery stats that gives an approximate estimation over a period of battery consumption. The authors in [43] adapted 17 sentence-level sentiment analysis methods of running on Android OS and then measured their memory, CPU usage and battery consumption efficiency. However, they correlated the battery consumption directly with the CPU usage rather than the actual measurement. The work in [31] describes the energy estimation of various mobile applications based on program level analysis, which uses a code prediction based approach.

It is evident that the existing works related to determining the energy consumption of a running process in mobile devices are mostly software-based (like powerstat,[5] PowerTutor[6] and others.) and hence not very accurate. A few works in the literature [46,8] have used hardware-based approaches for power measurement analysis of Internet access activities over mobile phones, which are known to be more accurate. In this work, we have adopted the hardware-based approaches, and also augmented the approaches for algorithmic and instruction-level power analysis.

3. Extractive text summarization

In this section, we first give an overview of some generic steps followed by extractive text summarization algorithms [14,24,30]. We next discuss the specific techniques followed by some text summarization algorithms that have been used in this paper for evaluation (we call them as base summarization algorithms).

3.1. Generic steps in extractive summarization

Analyzing a large number of extractive text summarization algorithms [14,24,30], we observe that such algorithms usually follows four generic steps – (1) pre-processing of tokens, (2) token-level metric calculation, (3) sentence scoring, and (4) final summary generation. A brief overview of these steps is discussed next.

(1) Pre-processing of the input text: Preprocessing involves sentence identification, words identification, removal of stop-words, and language-specific stemming of words.

(2) Token level metric calculation: After the preprocessing phase, significant tokens or words are identified based on their frequency or some other derivatives, like the probability of occurrence, among others. This step primarily involves calculating the *Term Frequency* (tf) and *Inverse Document Frequency* (idf) of the terms.

(3) Sentence scoring: In this step, scoring of all sentences of the entire input corpus (set of sentences) is done. Here, different algorithms use different approaches to handle tokens. While some algorithms opt for matrix-based operations (like Lex-Rank and LSA), others generally score sentences based on the significance of the terms contained in the sentences.

(4) Summary generation: The last step involves extracting the top-scoring sentences to provide a summary of suitable length.

3.2. Base summarization algorithms

We consider six classical extractive text summarization algorithms widely used in various applications [34,35,63,64], as the base summarization approaches. Note that we do not consider recent neural summarization models [18] in this study since their large energy consumption make them unsuitable for execution on mobile devices [33]. The extractive summarization algorithms that we consider as base algorithms are the following:

Frequency Summarizer (FS) [49]: This algorithm computes the frequency of each distinct token that is not a stop-word and discards all such tokens that are not in the allowable range of frequency. The sentences are then scored based on the presence of significant tokens and the sentences with the highest score are selected.

Luhn [40]: This approach is based on Luhn's classical method [40] that uses the `tf-idf` statistics to find the most frequent terms in a given document. This approach assumes that the significant words carrying most meaningful information are neither too frequent nor too seldom in the text.

Lex-Rank (LR) [20]: LR uses an unsupervised approach for text summarization, inspired by approaches PageRank [51] and HITS [45].

[5] https://www.hecticgeek.com/2012/02/powerstat-power-calculator-ubuntu-linux/.

[6] http://ziyang.eecs.umich.edu/projects/powertutor/.

For each phrase in the document, a graph is built here by generating a vertex. The edges between phrases are based on some sort of semantic similarity. Lex-Rank (LR) uses `Cosine_similarity` and `Power_method` for sentence scoring, by using the tf-idf vectors.

Latent Semantic Analysis (LSA) [59]: LSA uses singular value decomposition (SVD) for scoring the sentences. If a word combination pattern is dominant and recurrent in the document, one of the singular vectors captures this pattern and represents it. Here a matrix describes an importance degree of each topic in each sentence. The summarization process chooses the most informative sentence for each topic.

SumBasic (SB) [61]: SB is based on the word-probability distribution, where the input corpus is tokenized to obtain the highly probable tokens to be present in the summary. It also ensures that redundant tokens do not recur in the summary by removing each sentence and redefining the word probabilities.

COntent Word based Tweet Summarization (COWTS) [55]: This algorithm is specifically meant for summarizing tweets (from Twitter). It checks for nouns, verbs, emoticons and uses an abbreviation dictionary, which is done using the module `POS-Tagger`. It solves an Integer Linear Programming optimization problem to optimize the presence of important tokens in the summary. It can be noted that COWTS uses Gurobi optimizer which is not supported over ARM architecture that is used in a majority of mobile devices. To resolve this problem, we use *GNU Linear Programming Kit* (GLPK) optimizer[7] instead of Gurobi optimizer as a solver which can work over *ARM architecture*.

The algorithms mentioned above use different modules. Importantly, all the various modules can be mapped to one of the four generic steps described earlier in Section 3.1, as shown in Table 1. Later in this paper, we will perform a fine-grained analysis of the energy consumption of different modules of each algorithm, when executed over mobile devices. The next section describes the details of the experimental setup used to this end.

4. Experimental setup

This section details the devices and datasets used for experiments, followed by a description of the experimental procedure.

4.1. Mobile devices used

In developing countries, people mostly stick to buying budget to mid-range smartphones.[8] Since these low-cost smartphones have limited processing and battery power, running text summarization is a challenging task. Also, considering the emergency cases, e.g., disasters, we were motivated to devise a low energy-consuming summarization approach suited for low-cost mobile devices. Thus, we consider two different mobile phones for the experiments – one is a low-end smartphone (Samsung Galaxy S3 I9300), and the other is a mid-end smartphone (Gionee F103). We did not consider a high-end smartphone because High-end smartphones offer colossal computing power and good battery power life. Thus, immense computation is not a bottleneck for high-end smartphones. Also, today's high-end smartphones have embedded battery which are not suitable for connecting with the Monsoon Power monitor [57] discussed in Section 4.3.

The Samsung phone has 1 GB RAM with Exynox 4412 Quad Chipset, Quad-core 1.4 GHz Cortex-A9 CPU, and a 2100 mAH battery. Whereas, the mid-end Gionee phone has 2 GB of RAM with Mediatek MT6735 Chipset, Quad-core 1.3 GHz Cortex-A53 chipset, and a 2400 mAH battery. The Samsung phone is built using Lineage OS 14.1 on Android 7.1.2 operating system, whereas the Gionee phone is built over Amigo UI 3 on Android 6.0.1 operating system. The experiments have been performed keeping the battery charge of the phones in between 80% and 90%.

4.2. Datasets used

We use two different types of datasets – (i) datasets of microblogs/tweets crawled from Twitter, and (ii) standard DUC datasets that have been extensively used in the literature to analyze the performance of the various summarization approaches (details in Table 2).

Tweet Datasets: Motivated by the need for on-device summarization during disaster events, we collected relevant tweets posted during four disaster events – Typhoon Hagupit in 2014 ("Hgpt"), serial bomb blasts in Hyderabad, India in 2013 ("Hydb"), devastating floods in the Uttarakhand state of India in 2013 ("Utkd"), and 2017 Hurricane Irma in the Atlantic region ("Irma"). Relevant tweets were collected using keyword-based search through the Twitter API, e.g., the keywords 'Hagupit', 'storm' and typhoon' are used to identify tweets related to the Hagupit typhoon. Since tweets often contain duplicates and near-duplicates (e.g., retweets of the same tweet by many users), we removed them using the techniques suggested in [60]. Also, since tweets can consist of multiple sentences, we split those tweets using nltk[9] sentence-tokenizer. After sentence tokenization, we consider the four datasets, as stated in Table 2.

The standard procedure for evaluating the quality of summaries generated by an algorithm is to compare the algorithmically generated summaries with gold-standard summaries prepared by human annotators. Following the approach in [55], we employed four human volunteers (students from our institutes) for writing the gold standard summaries.[10] The annotators are well conversant with English and the use of social media. However, none of them is an author of this work. Each of them was asked to independently (i.e., not consulting one another) read the distinct sentences and prepare extractive summaries of length 20 sentences.

DUC Datasets: We have also conducted experiments over datasets provided by Data Understanding Conference (DUC) 2007, which have been used extensively as a benchmark for document summarization. The dataset contains 45 tasks/topics, where each task has 25 news articles (documents) from different news media sources, and 4 human-written summaries of 250 words each. We have analyzed all 45 topics in the DUC dataset in our work. However, for brevity in reporting results, we have randomly chosen five topics having various numbers of sentences, as shown in Table 2.

4.3. Energy Measurement Tool

To measure the energy consumption accurately at a level of algorithmic executions, we have used *Monsoon Power Monitor* [57], which is a hardware-based tool and provides a robust power measurement solution for mobile devices. Wires are connected to the battery terminals of the smartphone[11] to take instantaneous power readings. So, the smartphone battery must be removable. Thus, this technique is not suitable for smartphones having embedded batteries. The *Monsoon Power Monitor* outputs a discrete set of time versus instantaneous power consumption value [4,46,8]. Table 3 describes the system level and hardware specifications of the *Monsoon power monitor* along with prerequisite software and interfaces. Fig. 1 explains the details of our experimental setup. Next, we discuss the methodology used in our system to measure module-wise power consumption for various summarization approaches over mobile devices.

[7] https://www.gnu.org/software/glpk/.

[8] https://timesofindia.indiatimes.com/gadgets-news/this-is-how-much-indians-mostly-spend-on-smartphones/articleshow/70666576.cms.

[9] https://www.nltk.org/.

[10] The tweet datasets and the gold summaries can be found at https://github.com/Krishnandu91/Sustainable-Text-Summarization-over-Mobile-Devices-An-Energy-Aware-Approach.

[11] http://msoon.github.io/powermonitor/PowerTool/doc/Power%20Monitor%20Manual.pdf.

Table 1
List of modules of different base summarization approaches based on the generic steps.

Summarization approaches	Generic summarization steps.			
	Pre-processing of tokens	Token level metric calculation	Sentence scoring	Generate summary
Frequency Summarizer (FS) [49]	`init`	`Compute_tf`	`Compute_rank`	
Luhn [40]	`Get_significant_words`		`Rate_sentences`	
Lex-Rank (LR) [20]	`To_words_set`	`Compute_tf`, `Compute_idf`	`Cosine_similarity`, `Power_method`	`Select_top_ranked_sentences`
Latent Semantic Analysis (LSA) [59]	`Create_dictionary`, `Create_matrix`	`Compute_tf`	`SVD`, `Compute_Rank`	
SumBasic (SB) [61]	`Get_probabilities`, `Get_sentences`		`Max_sentences`	
COWTS [55]	`POS-Tagger`	`Compute_tf`	`ILP solving`	

Table 2
Description of tweet datasets and DUC datasets.

Dataset	Description	Number of sentences	Number of words
Hgpt	"Typhoon Hagupit" in Philippines (2014)	500	8192
Hybd	"Bomb blasts" in Hyderabad, India (2013)	1000	15,492
Utkd	Floods and landslides in Uttarakhand, India (2013)	2000	31,548
Irma	"Hurricane Irma" in the Atlantic region (2017)	10,000	125,482
D1	DUC 2007 (Task 16)	283	7185
D2	DUC 2007 (Task 17)	546	11,545
D3	DUC 2007 (Task 18)	625	12,384
D4	DUC 2007 (Task 36)	654	11,941
D5	DUC 2007 (Task 33)	791	17,374

Table 3
Monsoon power monitor specifications.

Model	LVPM, Part Number FTA22D
Legacy s/w version	PowerTool 4.x
OS requirements	Windows Vista or higher
Interface requirement	Full speed USB 1.1/2.0/3.0

4.3.1. *Module-wise energy estimation technique using power monitor*

Monsoon power monitor provides instantaneous power consumption for each time-instance while a summarization task is executed over the mobile device. The readings provide a set of instantaneous power consumption for the summarization task as a whole. Thus to analyze the module-wise energy consumption of a summarization approach, we introduce sleep states (idle states) before and after the execution of each module given in Table 1. These sleep states help to segregate the modules from each other and break down the readings of the power monitor into separate logical chunks. As an example, Fig. 2 shows how the modular analysis is done by introducing the sleep states in between the modular executions of the *LSA* summarization approach. This helps us to extract the power consumption for different modules. If the sleep states were not introduced in between modules, the algorithm would have run as a whole, and modular energy analysis would not have been possible. Once the energy consumption analysis is done, the idle states are removed.

Using the power monitor, we take readings for two different states – *Idle state* and *Modular state*, while executing a summarization approach by introducing the sleep states. The *Idle state* refers to the time when no module of the summarization approach is executed (sleep state), however, there may be some daemon (small background) processes running over the mobile devices. On the other hand, the *Modular state* refers to the time when a module of the summarization approach is being executed.

Algorithm 1. Calculation of energy using power monitor.

Input: Idle state instantaneous power consumption samples (Time,Power) and Modular state instantaneous power consumption samples (Time,Power).
Output: Modular energy consumption.
procedure ESTIMATE-ENERGY
  Read distinct pairs of (time, power) for idle state samples **(ISS)**
  N = Length(ISS)
  Threshold(t) = 0
  for each ($time_i$, $power_i$)∈ (ISS) **do**
    $t = t + power_i$
  $t = t/N$
  Read pairs of (time, power) for modular state samples **(MSS)**
  for each ($time_i$, $power_i$)∈ (MSS) **do**
    if $power_i > t$ **then**
      x[].append($time_i$)
      y[].append($power_i$)
  Use Simpson's rule for finding area under the curve
  Area, A = simpson(x,y)
  return A

Algorithm 1 describes the procedure that we follow to calculate the total energy consumption by measuring the area under the curve for the energy consumption of a module (`SVD`) of the *LSA* summarization approach (shaded region in Fig. 2). First, we need to purge the power consumption by the background daemon processes. For this, we calculate a threshold power value by measuring power consumption at the idle states at different time instances, called the *Idle State Samples* (*ISS*). This gives the power consumption by the background daemon processes. Thus it can be replaced with an average value that determines a threshold, above which we can safely conclude the amount of energy consumption while running a process other than the background processes. Then, the total energy is calculated by taking *Modular State Samples* (*MSS*), the periodic sample measurements of power consumption during the execution of a module, and the threshold value calculated and then using these values to obtain the *area under the curve* by using Simpson's Integral rule.[12] This gives the total power consumption by a module of a summarization approach.

4.4. *Measuring summary quality*

Apart from power consumption, we also look into the summarization quality for different summarization approaches. To this end, we use the standard ROUGE metric [25] that measures the overlap of an algorithm-generated summary with one or more human-generated summaries (called gold standard summaries). In our work, we have experimented with ROUGE-2 and ROUGE-SU4 metrics. ROUGE-2 measures the overlap of *bigrams*, and ROUGE-SU4 measures the overlap of bigrams with a maximum skip distance of 4 between bigrams, in terms of *Precision* and *Recall*. *Recall* measures what fraction of gold standard summary is included in the algorithm-generated summary and *Precision*

[12] http://mathworld.wolfram.com/SimpsonsRule.html.

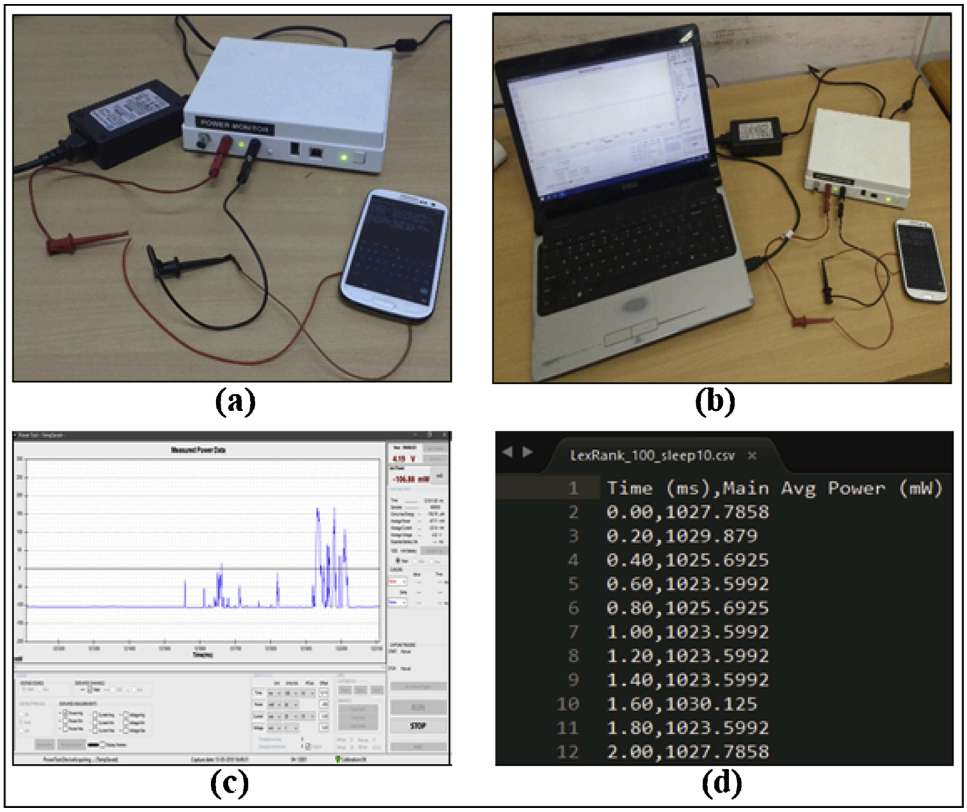


Fig. 1. Experimental setup: (a) interface between *Monsoon power monitor* and Mobile device, (b) interface between the *Monsoon power monitor* and Computer, (c) software interface of *Monsoon power monitor*, (d) *Monsoon power monitor* readings for instantaneous power consumption.

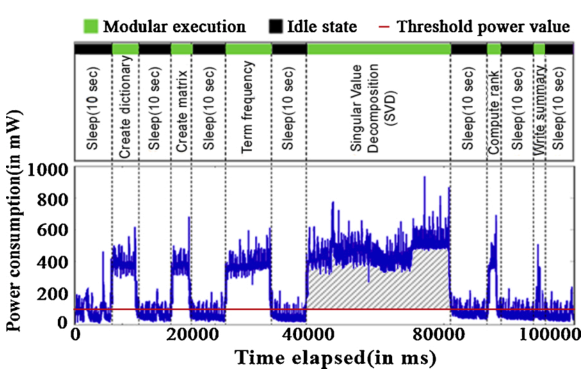


Fig. 2. Instantaneous power consumption of modules of LSA summarization approach along with introduced sleep states while running in a mobile device.

indicates what fraction of the algorithm-generated summary is contained in the gold-standard summary. We considered the F_Score which is the harmonic mean of *Precision* and *Recall*. From Table 4, we can observe that even though the F_Scores for ROUGE-2 and ROUGE-SU4 are different, the trend over various algorithms is the same. The F_Scores for ROUGE-SU4 are higher than the corresponding ROUGE-2 F_Scores. However, the relative difference of F_Scores between the algorithms is similar for ROUGE-2 and ROUGE-SU4. Thus for the sake of simplicity, we have presented only ROUGE-2 F_Scores after that.

5. Energy consumption vs. summary quality: a 1000-foot view

In this section, we discuss the overall performance of the base summarization algorithms (stated in Section 3.2) over the two mobile devices considered. We look into three performance metrics – energy consumption, summary quality and execution time. Unless otherwise mentioned, every summarization algorithm is made to generate a summary of 20 sentences for each of the dataset, irrespective of the dataset size. We give the same input to all the summarization approaches; however, different summarization approaches may generate a different summary of size 20 sentences.

5.1. Analysis of energy consumption

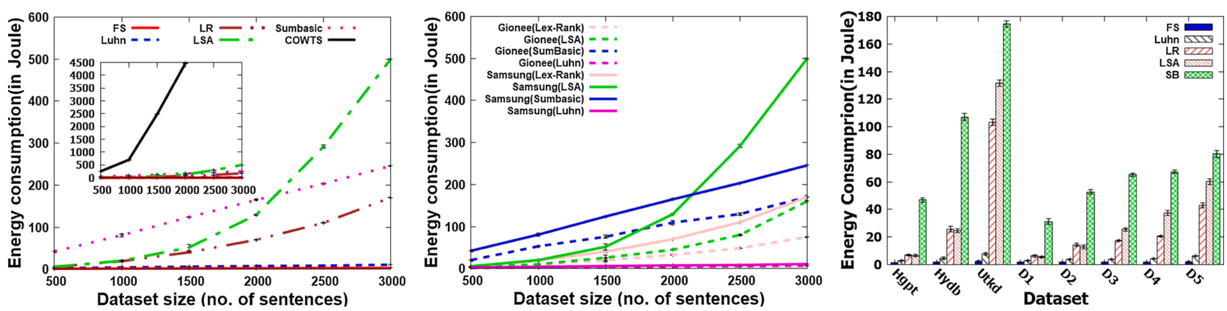
First, we look into the total energy consumption of various summarization algorithms with respect to the increased dataset size, using the Irma tweet dataset. We vary the input data size (number of sentences) from 500 to 3000, and for each case, we produce a summary of 20 sentences by each algorithm. We execute each of the sizes of the above experiment for 5 times with randomly selected sentences of the given input size from the entire dataset. The energy consumption characteristics for the six summarization algorithms are shown in Fig. 3(a). We observe that the energy consumption of COWTS increases exponentially with respect to the input data size. Also, the energy consumption for LSA and LR are significantly higher, while FS and Luhn are the most energy-efficient compared to others.

Next, we observe the impact of device specifications on the energy consumption for various summarization algorithms (we do not consider COWTS in this experiment as it consumes very high energy). Fig. 3(b) shows that the energy consumption of various summarization algorithms varies based on the device capacity, but *the trends for the energy consumption behaviour of all summarization algorithms remain similar across different devices*. In the above two Fig. 3(a),3 (b) we have considered only Irma (tweet dataset) because it has an exceptionally high number of sentences(please refer to Table 2). The comparison reveals how the summarization approaches become inefficient over a large dataset. Thus, the other datasets having small dataset sizes are not suitable for the purpose.

Finally, we check the impact of input data type on the energy

Table 4
Quality of summary of the base summarization algorithms in terms of ROUGE-2 and ROUGE-SU4 F_Scores; light green (bold) and light blue (italic) cells respectively represent the maximum and the second-maximum score for a dataset.

Dataset	ROUGE	Algorithms					
		FS	Luhn	LR	LSA	SB	COWTS
Hgpt	ROUGE-2	0.187	0.21	0.32	*0.34*	0.24	***0.40***
	ROUGE-SU4	0.21	0.26	0.37	*0.38*	0.27	**0.45**
Hybd	ROUGE-2	0.19	0.22	*0.33*	0.30	0.22	**0.45**
	ROUGE-SU4	0.23	0.25	*0.35*	0.32	0.26	**0.47**
Utkd	ROUGE-2	0.209	0.21	0.31	*0.32*	0.20	**0.42**
	ROUGE-SU4	0.21	0.22	0.33	*0.33*	0.22	**0.44**
D1	ROUGE-2	0.101	0.109	**0.138**	*0.135*	0.065	0.096
	ROUGE-SU4	0.123	0.134	**0.181**	*0.175*	0.090	0.119
D2	ROUGE-2	0.061	0.111	**0.157**	*0.125*	0.108	0.118
	ROUGE-SU4	0.092	0.144	**0.201**	*0.160*	0.133	0.145
D3	ROUGE-2	0.059	0.062	*0.079*	**0.080**	0.051	0.039
	ROUGE-SU4	0.091	0.092	*0.101*	**0.103**	0.088	0.043
D4	ROUGE-2	0.027	0.039	**0.077**	0.051	0.034	*0.071*
	ROUGE-SU4	0.081	0.099	**0.132**	0.101	0.082	*0.120*
D5	ROUGE-2	0.048	0.080	**0.132**	*0.098*	0.076	0.038
	ROUGE-SU4	0.092	0.139	**0.199**	*0.150*	0.132	0.080



(a) (b) (c)

Fig. 3. (a) Energy consumption versus dataset size (Inset: energy consumption of COWTS), (b) energy consumption of base summarization approaches on different mobile devices, (c) energy consumption of base summarization approaches over different datasets.

consumption behaviour of the summarization approaches. We have two different types of data – tweet data and text documents (DUC datasets). Fig. 3(c) shows the average energy consumption for various summarization algorithms over different datasets. The tweet datasets, in general, require more energy than the DUC datasets for the majority of the summarization algorithms, primarily because the tweet datasets have more number of words compared to the DUC datasets. However, interestingly, the LSA algorithm consumes more energy for D4 dataset compared to the Hydb tweet dataset, although the number of words in Hydb is more than that of D4 (check Table 2). By close inspection, we observe that the major processing modules of LSA depend on the number of *distinct* words in the dataset, rather than the total number of words; therefore, the energy consumption performance of LSA depends on the number of distinct words in the input data.

5.2. Analysis of summary quality

Next, we observe that the *summary quality is independent of the mobile device used*, and therefore, we report the average summary quality over the two devices. The ROUGE-2 and ROUGE-SU4 F_Score for different summarization algorithms over different datasets is reported in Table 4. For the tweet datasets, COWTS performs much better than other base summarization algorithms, probably because COWTS is specifically designed for tweets posted during disaster situations. For the DUC datasets, LR and LSA achieve higher scores than SB, FS, Luhn, and COWTS, probably because LR and LSA use matrix-based methods to capture semantically similar sentences.

5.3. Analysis of execution time

Finally, we look into the overall execution time of various summarization approaches when executed over mobile devices. Tables 5 and 6 show the execution time (in seconds) required to execute the considered base summarization approaches over Samsung Galaxy S3 I9300 and Gionee F103 respectively for different datasets. The execution time shows a similar pattern for both devices. We observe that the execution time is mostly correlated with the energy consumption behaviour, but there are exceptions, e.g., SB, which takes less amount of time to execute but consumes a relatively high amount of energy compared to LR, Luhn, or LSA. We have noticed that SB involves energy-hungry computations for finding out the highly probable tokens, which require frequent access and operations over the tokens (module-wise analysis has been

Table 5
Execution time (in seconds) of the base summarization approaches; light green and light blue cells respectively represent maximum and second maximum execution time of the summarization approaches for a dataset running in Samsung GalaxyS3 I9300.

Dataset	Summarization approaches					
	FS	LR	Luhn	LSA	SB	COWTS
Hgpt	5.03	20.2	8.32	17.87	8.23	948.36
Hydb	6.2	41.81	13.55	39.64	10.25	2077.1
Utkd	8.2	192.35	22.41	298.66	25.13	10208.2
D1	5.25	18.04	8.75	16.6	6.02	509.67
D2	5.75	38.89	11.04	34.05	7.85	1175.5
D3	6.01	51.63	12.89	70.16	8.41	1763.89
D4	6.09	53.07	13.37	84.32	8.74	1855.32
D5	6.94	75.5	15.36	119.07	10.87	3545.68

Table 6
Execution time (in seconds) for base summarization approaches; light green and light blue cells respectively represent maximum and second maximum execution time of the summarization approaches for a dataset running in Gionee F103.

Dataset	Summarization approaches					
	FS	LR	Luhn	LSA	SB	COWTS
Hgpt	5.78	20.29	8.8	14.1	7.87	905.39
Hydb	6.43	49.03	13.36	36.91	9.15	2025.45
Utkd	8.9	259.53	28.86	276.97	23.45	10168.7
D1	5.37	18.14	7.98	14.36	5.46	501.54
D2	5.98	41.22	10.36	31.56	7.11	1137.36
D3	6.2	54.37	12.6	63.48	7.98	1734.78
D4	6.95	56.1	12.91	81.76	8.3	1814.39
D5	7.06	81.48	16.29	116.2	10.62	3501.65

discussed later). This increases the instantaneous power consumption at every time-instances although the total execution time is less. On the other hand, we observe that FS has a similar execution time as of SB; however, FS consumes less energy compared to SB. FS directly computes the frequency of the tokens rather than computing the probability of occurrences for each token, and the operations require less instantaneous power consumption. From these observations, we can conclude that we cannot determine the power consumption of a summarization approach only by considering the overall execution time because it also depends on the methodologies used.

5.4. Looking ahead

From the above analyses, we have a few important observations.

- COWTS consumes very high energy; hence, it is not suitable for execution over mobile devices. On the other hand, the SB algorithm is neither energy-efficient nor produces a good quality summary. As a consequence, we do not consider these two algorithms for further analysis.
- There is a trade-off between energy consumption and summary quality. The summarization approaches (like LR and LSA) that generally produce better quality summaries consume more energy than others.
- Energy consumption of summarization algorithms is dependent on the particular mobile device, but the consumption patterns remain the same across different devices.
- Energy consumption varies proportionally to the input data size in terms of the total number of words or number of distinct words.
- Execution time is not directly correlated with the energy consumption for summarization algorithms; therefore, we cannot consider execution time as a direct metric to develop an energy-efficient summarization approach.

These observations motivate an in-depth analysis of the energy consumption of various modules used in different algorithms. We observed earlier that four generic steps are followed by all the summarization algorithms, and different techniques are used in those steps by various summarization algorithms. We next proceed to figure out the energy consumption behaviour of those individual techniques or modules used in the four steps.

6. Deep dive towards energy-efficient summarization

We next explore the energy-consumption behaviour of different modules used in the four steps of the summarization approaches (see Table 1), aiming to develop an energy-efficient summarization algorithm. It can be noted that every module uses a well-defined algorithm to perform a given task like token generation, token level metric calculation, sentence scoring, and others. Therefore, we first look into module-level energy consumption, and then design an energy-efficient approach for summarization.

6.1. Module-wise energy consumption analysis

Here we analyze the energy consumption for various modules under the four generic steps given in Table 1 in Section 3. We have followed the approach as discussed in Section 4.3.1. The average energy consumption for various modules has been summarized in Table 7, when a summarization approach is executed with a given dataset. It can be noted that the `Compute_rank` modules for FS and LSA use a different strategy for ranking the sentences; therefore, we have measured and reported the energy consumption separately for these two cases.

From Table 7, we observe that the energy consumption for the modules in the first (*pre-processing of tokens*) and the second steps (*token level metric calculation*) do not consume significant energy. Further, module-wise energy variation for these two steps is very less. Therefore, the summarization methods do not differ much in terms of energy consumption considering the first, second and fourth steps (the fourth step is similar for all the summarization approaches). However, the energy consumption for the modules in the third step (*sentence scoring*) varies a lot; while `Power_method` consumes around 0.1–0.3 Joule energy across all the datasets, `SVD` consumes as high as 121.1 Joule energy for the Utkd dataset. Therefore, we need to focus on the third step while designing a power-efficient summarization technique.

6.2. Energy-efficient hybrid summarization

Our approach for designing energy-efficient summarization for mobile devices is based on the following observations.

- From Table 7, among the four steps, the sentence scoring step has maximum effect on the energy consumption of summarization algorithms.
- From Table 4, we observe LSA and LR produce better quality summaries than FS and Luhn, irrespective of the dataset. However, in Table 7, we observe the modules used for sentence scoring in LSA and LR consume a significant amount of energy, and the energy consumption increases rapidly with the increase in the dataset size.
- Although from Table 4, we observe the quality of summary produced by FS and Luhn is poor, in Table 7, we observe the sentence scoring modules of FS and Luhn consume very less energy; the increment of energy consumption with the increase in dataset size is insignificant.

The above observations motivate us to design a hybrid summarization mechanism by combining these two classes of summarization approaches – reduce the input data size by generating an intermediate summary with the help of the sentence scoring algorithm used in FS or Luhn, and then use the sentence scoring algorithm in LSA and LR for

Table 7
Energy Consumption of the basic modules grouped by generic steps; For each of the four generic steps, the light green (bold) and the light blue (italic) cells represent the maximum and the second maximum energy consumption, respectively, for each dataset, among the modules used in a generic step.

Logical modules		Energy consumed (J)							
		Dataset							
		Hgpt 500	Hydb 1000	Utkd 2000	D1 283	D2 546	D3 625	D4 654	D5 791
Pre-processing of tokens									
`To_word_set`		**2.2**	*2.3*	3.3	**1.6**	**1.6**	**1.7**	**1.6**	**2.1**
`Create_dictionary`		*2*	**2.5**	*3.7*	*1.5*	*1.4*	*1.5*	*1.4*	*1.5*
`Create_matrix`		0.9	1.2	**4.3**	0.7	1	1.1	1.1	1.3
Token level metric calculation									
`Compute_tf`		*0.5*	0.6	0.9	*0.4*	0.6	0.6	0.7	1
`Compute_idf`		0.2	*1*	*3.7*	0.2	*0.8*	*1.2*	*1.6*	**3.3**
`Get_signifcant_words`		**1.9**	**2.5**	**3.9**	**1.8**	**2.1**	**2.3**	**2.7**	*2.8*
Sentence scoring									
`Power_method`		0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.2
`Compute_rank`	`FS`	0.4	0.5	0.8	0.0	0.2	0.2	0.2	0.3
	`LSA`	0.1	0.3	0.9	0.1	0.3	0.9	0.8	1.5
`Rate_sentences`		0.5	1.5	3.3	0.6	1.1	1.2	1.2	2.6
`Cosine_similarity`		**3.4**	**21.2**	*94.5*	**8.5**	**26.1**	**27.5**	**24.5**	**35.6**
`SVD`		*2.5*	*19.4*	**121.1**	*1.9*	*10.2*	*15.4*	*17.6*	*28.2*
Generate Summary									
`Write_summary`		0.5	0.5	0.7	0.75	0.6	0.5	0.6	0.6

producing good quality final summary from the intermediate summary. From these combinations, we propose four different variants of hybrid summarization mechanisms by combining the sentence scoring approaches in two steps, as shown in Fig. 4 – **(1) FSLSA** (FS `Compute_rank` followed by LSA `SVD` and `Compute_rank`), **(2) FSLEX** (FS `Compute_rank` followed by LR `Cosine_similarity` and `Power_method`), **(3) LULSA** (Luhn `Rate_sentences` followed by LSA `SVD` and `Compute_rank`) and **(4) LULEX** (Luhn `Rate_sentences` followed by `Cosine_similarity` and `Power_method` of LR). Next, we discuss the performance of these four variants of hybrid summarization along with the impact of different configuration parameters.

7. Performance of hybrid summarization approaches

We analyze the energy consumption and the quality of summary for proposed hybrid approaches with respect to varying the number of sentences in the intermediate summary. We represent the number of sentences in the intermediate summary as L. Table 8 shows the results we obtain from the proposed hybrid energy-aware summarization approaches. It describes how the energy consumption and the quality of summary vary with respect to the value of L. The analysis in Table 8 is based on a tweet dataset (Utkd) and a DUC dataset (D5). The input corpus size is 2000 and 791 sentences for the tweet dataset and DUC dataset, respectively. The final output size is 20 sentences. The value of L is taken by doubling the summary length each time, starting from 20 sentences. Table 9 depicts the performance of the base summarization approaches when the summarization is done in one go on a tweet dataset (Utkd) and a DUC dataset (D5). From comparing Tables 8 and 9, we observe that the proposed hybrid summarization approaches consume a fraction of the energy consumed by the computationally expensive base summarization approaches (LSA, LR) while providing a significantly higher quality of summary compared to the low energy-consuming base summarization approaches (FS, Luhn).

From Table 8 we also observe that as we increase the value of L, the energy consumption increases. Thus, there arises a need to define a suitable value of L so that the summarization can be executed in a given energy budget.

7.1. Selecting intermediate summary length

We describe an empirical method by performing experiments to choose the value of L for a given energy budget so that the energy consumption is reduced without much compromising with the quality of the summary. We take LULEX as an example. Using the values from Table 8 for the tweet dataset (Utkd), we plot the graph shown in Fig. 5.

In this example, let us consider that the energy budget available is 60 J. From Fig. 5, we can identify that 910 sentences is the maximum possible value of L, i.e., the *maximum intermediate summary length* for the given energy budget for the said summarization approach. Since the energy consumption increases with an increase in the value of L, thus any lower value will naturally consume less energy than specified in a given energy budget.

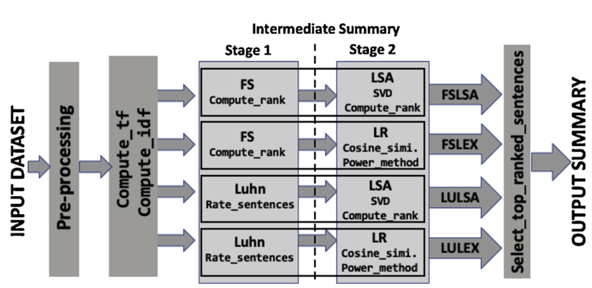


Fig. 4. Two step hybrid summarization framework containing the sentence scoring for four approaches.

Now, when we map the obtained values of L in Fig. 6 for the same hybrid summarization approach, we get a set of values for the quality of summaries as the probable candidates. Suppose, we denote this entire set of values by S. Then the best value will be $max(S)$ and the corresponding value in the X-axis will be the ideal value of L. In this example, from Fig. 6, we observe 640 sentences to be the ideal value of L.

7.2. Performance comparison: hybrid vs. baseline summarization approaches

We observed from Tables 4 and 9 that the quality of summary produced by the low-performing summarization approaches (FS, Luhn) is very poor. Thus, we compare our proposed hybrid summarization approaches with the high-performing baseline summarization approaches (LSA, LR) considering parameters – *Energy improvement* and *Quality degradation* with a successive increase in the energy budget. We define the performance parameters as,

$$\text{Energy Improvement } (\%) = \frac{E_{BA} - E_{HA}}{E_{BA}} \times 100 \tag{1}$$

$$\text{Quality Degradation } (\%) = \frac{Q_{BA} - Q_{HA}}{Q_{BA}} \times 100 \tag{2}$$

where E_{BA}, E_{HA}, Q_{BA} and Q_{HA} stands for the *energy consumption for baseline summarization approach*, the *energy consumption for hybrid summarization approach*, the *quality of summary for baseline summarization approach* and the *quality of summary for hybrid summarization approach*, respectively.

Previously, we have shown the results for the energy consumption and the quality of summary obtained from LULEX hybrid summarization with a given energy budget and thus obtained the suitable intermediate summary length. Now, to obtain a statistical overview of all four hybrid summarization approaches, we conduct the experiment as mentioned in Section 7.1 taking different energy budgets from 10 to 50 J.

In Table 11, the performance of each hybrid summarization approach is measured with respect to the baseline summarization approaches (LSA, LR). We observe that the hybrid summarization approaches exhibit a significant improvement of 51–95% in energy consumption, where the degradation of quality is 5–29% over the high-performing baseline summarization approaches in the case of tweet dataset (Utkd). Similarly, for the DUC dataset (D5), the hybrid summarization approaches show an improvement of 64–90% in energy consumption where the degradation of quality is 5–42% over the baseline summarization approaches.

Next to determine the sustainability of our proposed hybrid summarization approach we consider to run the hybrid summarization approach on a very large dataset.

7.3. Energy consumption of hybrid summarization approaches for large dataset

We further explore the potential of the hybrid summarization approaches in reducing the energy consumption, assuming the quality of summary to show a similar trend as the previous experiments. We have applied the summarization on a very large dataset of 10,000 sentences (Irma). In each case, we have taken the number of sentences in the intermediate summary half the number of sentences in the total dataset. From Table 10, it is apparent that the energy consumption of hybrid summarization approaches is much lower than their respective high-performance summarization approaches (LSA, LR).

From Table 10, we can observe the energy improvement(%) in energy consumption by the hybrid summarization approaches with respect to their corresponding high-performing baseline summarization approaches. The energy improvement(%) varies from 81% to 89% in the case of FSLSA and LULSA, and from 66% to 76% in the case of FSLEX and

Table 8
Energy consumption and quality of summary of the proposed four hybrid summarization approaches with respect to the variable size of the intermediate summary; light yellow (bold) and light green (italic) cells respectively represent the minimum and the second minimum energy consumption, whereas the light red (bold) and light blue (italic) cells respectively represent the maximum and the second maximum quality among the proposed hybrid approaches.

Intermediate summary size (L)	FSLSA		FSLEX		LULSA		LULEX	
	Energy (J)	R2 (F_score)	Energy(J)	R2 (F_score)	Energy(J)	R2 (F_score)	Energy(J)	R2 (F_score)
Hybrid summarization approaches (running over Utkd tweet dataset)								
40	**4.4**	*0.226*	*4.6*	0.221	9.4	0.222	10.4	**0.246**
80	**4.6**	*0.243*	*4.7*	0.241	11	**0.246**	11.3	0.228
160	**5**	**0.265**	*5.3*	0.236	11.8	0.241	13.9	*0.262*
320	**6.4**	0.282	*6.7*	0.255	19.3	**0.286**	20.6	*0.284*
640	*15.9*	*0.286*	**14.4**	0.283	46.9	0.261	40.8	**0.293**
1280	*38.3*	0.298	**32.2**	0.301	127.9	**0.312**	80.4	*0.307*
Hybrid summarization approaches (running over D5 DUC dataset)								
40	**3.69**	0.038	*3.96*	0.07	10.6	*0.086*	11.24	**0.087**
80	**3.81**	0.049	*4.71*	0.073	11.48	**0.1**	12.18	*0.091*
160	*4.28*	0.046	**4.11**	0.066	12.28	*0.092*	13.98	**0.103**
320	*6.28*	0.075	**4.73**	0.078	36.24	*0.093*	30.18	**0.114**
640	*14.27*	0.078	**12.43**	0.081	53.58	*0.092*	56.12	**0.13**

Table 9
Energy consumption and quality of the summary of the four base summarization approaches with respect to the variable size of the intermediate summary. The bold values represent the maximum energy consumption and the maximum quality of summary of the base approaches in each row corresponding to each dataset.

Dataset	Base summarization approaches							
	FS		Luhn		LR		LSA	
	Energy (J)	R2 (F_score)	Energy (J)	R2 (F_score)	Energy (J)	R2 (F_score)	Energy (J)	R2 (F_score)
Utkd	2.46	0.209	7.9	0.212	103.4	0.315	**131.6**	**0.326**
D5	1.98	0.048	6.23	0.084	42.95	**0.132**	**60.12**	0.099

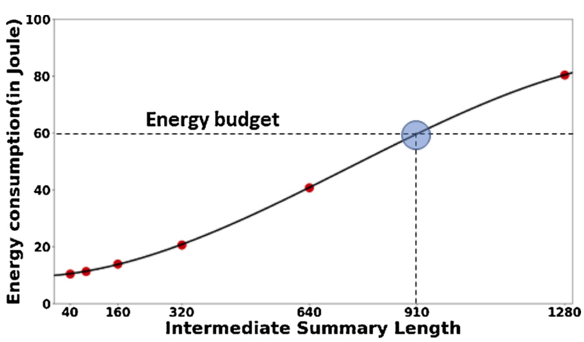


Fig. 5. Energy consumption of LULEX for different intermediate summary lengths. The dotted line depicts the energy budget of 60 J. The x-coordinate corresponding to the blue circle gives the maximum value of an intermediate summary for the given budget.

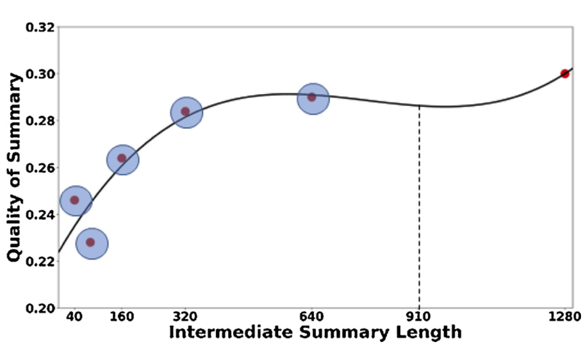


Fig. 6. Quality of summary for LULEX for different intermediate summary lengths. The encircled red dots denote the achievable quality of summary within the proposed budget described in Fig. 5.

LULEX. It can also be noticed that the improvement is more significant when the dataset size is large, e.g., we can save up to 21,000 J when we use the FSLSA or LULSA in-place of LSA when we summarize a dataset of 10,000 sentences.

The low-end device (Samsung GalaxyS3 I9300) has a standard 3.7 V battery of 2100 mAh capacity. Though the battery drainage is not always linear, we can roughly estimate the battery drainage percentage using Eqs. (3) and (4):

$$\text{Charge}\quad(\text{mAh}) = \frac{\text{Energy}\quad(\text{J})}{\text{Voltage}\quad(\text{V}) \times 3.6} \tag{3}$$

$$\text{Battery}\quad\text{Drainage}\quad(\%) = \frac{\text{Charge}\quad\text{Consumed}\quad(\text{mAh})}{\text{Total}\quad\text{Charge}\quad(\text{mAh})} \times 100 \tag{4}$$

From Table 10, we can observe LR, LSA consumes over 2000 J energy to summarize a dataset of 10,000 sentences. 2000 J energy consumption is equivalent to a 7–8% battery drop. In real-time, a user would possibly run the text summarizer numerous times as needed. Thus, in a crisis scenario running the text summarizer would not be possible. However, running any hybrid summarization approach with a specified energy budget, e.g., 60 J, is equivalent to only 0.25% of the total battery life. Thus, we can significantly improve the battery duration.

8. Conclusion and future scope

In this paper, we have studied the existing summarization approaches at a modular level. Then, we have analyzed the summarization approaches with respect to the energy consumption, the quality of summary, and the execution time on varying parameters like dataset size, dataset type, and mobile device used. For energy consumption measurement at a modular level, we have defined a strategy using *Monsoon power monitor*. From the observations we have obtained from the above analysis, we have been motivated to propose a hybrid framework. Thus, we have given a proof of concept of how we can minimize the energy consumption by around 50–90% with degradation of summary quality by nearly 5–40% based on the selection of different configuration parameters. We have shown how the hybrid summarization approaches can be amended by tuning the intermediate summary length while an energy budget is provided. Last but not least, we have shown that our proposed hybrid algorithms give a promising solution for developing a sustainable energy-aware summarization approach that

Table 10
Energy consumption (J) and Energy Improvement (%) of the hybrid approaches for large dataset size; light yellow cells represent energy consumption for LSA and the hybrids which use LSA; similarly, light green cells represent energy consumption for LR and the hybrids which use LR.

Dataset size (no. of sentences)	Energy consumption (J) and Energy Improvement(%)									
	LSA	FSLSA		LULSA		LR	FSLEX		LULEX	
	J	J	%	J	%	J	J	%	J	%
2000	**129.3**	21.96	*83*	24.53	*81*	**70.03**	21	*70*	23.57	*66.3*
4000	**1260.21**	131.7	*89.5*	136.77	*89.1*	**309.18**	72.43	*76.6*	77.5	*74.9*
6000	**4611.32**	503.13	*89*	510.82	*89*	**728.36**	173.49	*76.2*	181.18	*75.1*
8000	**11594.05**	1263.81	*89.1*	1273.34	*89*	**1335.36**	312.78	*76.6*	322.31	*75.6*
10000	**23703.75**	2573.82	*89.1*	2585.45	*89.1*	**2136.95**	500.33	*76.6*	511.96	*76*

Table 11
Performance of hybrid summarization approaches with respect to high-performing base summarization approaches(LSA, LR) over tweet (Utkd) and DUC (D5) datasets; light yellow (bold) and light green (italic) cells respectively represent maximum and second maximum energy improvement(%) @ quality degradation(%) among the proposed hybrid approaches for a given energy budget (J).

Hybrid summarization approaches (running over tweet dataset)					
	Energy Budget (J)	FSLEX	FSLSA	LULEX	LULSA
Energy Improvement (%) @ Quality Degradation (%)	10J	*90% @ 20%*	**95% @ 15%**	86% @ 25%	92% @ 29%
	20J	*86% @ 14%*	**92% @ 7%**	81% @ 16%	88% @ 20%
	30J	*81% @ 11%*	**88% @ 10%**	76% @ 12%	86% @ 14%
	40J	**76% @ 8%**	85% @ 12%	*70% @ 8%*	84% @ 16%
	50J	**72% @ 7%**	84% @ 14%	*64% @ 7%*	81% @ 17%
Hybrid summarization approaches (running over DUC dataset)					
Energy Improvement (%) @ Quality Degradation (%)	10J	*90% @ 33%*	87% @ 42%	**82% @ 6%**	74% @ 34%
	20J	*88% @ 24%*	87% @ 40%	**80% @ 4%**	70% @ 25%
	30J	*88% @ 24%*	87% @ 40%	**78% @ 5%**	66% @ 22%
	40J	*88% @ 24%*	87% @ 40%	**76% @ 5%**	64% @ 19%
	50J	*88% @ 24%*	87% @ 40%	**75% @ 5%**	64% @ 17%

minimizes energy efficiency while giving a good quality summary.

Thus, this work is the first of its kind, which can pave the way for further detailed research and analysis on aspects of energy consumption of various text summarization approaches at the device-level as well as the process-level. In the future, we want to generalize finding the *ideal intermediate summary*, which is highly dependent on the dataset. We would also like to study the effect on energy consumption by making code-level modifications, e.g., changing data structures and libraries of the text summarization modules. We also plan to extend the analysis for abstractive text summarization approaches, which are more complex for execution over the mobile devices.

Conflict of interest

None declared.

Acknowledgement

This work is an outcome of the R&D work undertaken in the project entitled "Developing ICT-based kiosks for post-disaster situational information management using opportunistic networking framework" funded by Department of Higher Education, Science & Technology and Biotechnology, Government of West Bengal, India.

References

[1] R.M. Alguliyev, R.M. Aliguliyev, N.R. Isazade, A. Abdi, N. Idris, Cosum: text summarization based on clustering and optimization, Expert Syst. 36 (2019) e12340.
[2] Y. Atoji, T. Koiso, M. Nakatani, S. Nishida, An information filtering method for emergency management, Electr. Eng. Jpn. 147 (2004) 60–69.
[3] H. Ba, W. Heinzelman, C.A. Janssen, J. Shi, Mobile computing – a green computing resource, 2013 IEEE Wireless Communications and Networking Conference (WCNC) (2013) 4451–4456.
[4] N. Balasubramanian, A. Balasubramanian, A. Venkataramani, Energy consumption in mobile phones: a measurement study and implications for network applications, Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement (2009).
[5] O. Buyukkokten, H. Garcia-Molina, A. Paepcke, Seeing the whole in parts: text summarization for web browsing on handheld devices, Proceedings of the 10th International Conference on World Wide Web (2001).
[6] L. Cabral, R. Lima, R. Lins, M. Neto, R. Ferreira, S. Simske, M. Riss, Automatic summarization of news articles in mobile devices, Proceedings of the 15th IEEE Mexican International Conference on Artificial Intelligence (2015).
[7] L. Cabral, M. Neto, A. Borges, R. Lins, R. Lima, R. Mello, M. Riss, S.J. Simske, Mobile summarizer and news summary navigator: Two multilingual news article summarization tools for mobile devices, Proceedings of the 2016 ACM Symposium on Document Engineering (2016) 107–110.
[8] Y. Cao, J. Nejati, M. Wajahat, A. Balasubramanian, A. Gandhi, Deconstructing the energy consumption of the mobile page load, Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 1 (2017) 6.
[9] J.Y. Chin, S.S. Bhowmick, A. Jatowt, Totem: personal tweets summarization on mobile devices, Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (2017) 1305–1308.
[10] J.Y. Chin, S.S. Bhowmick, A. Jatowt, On-demand recent personal tweets summarization on mobile devices, J. Assoc. Inform. Sci. Technol. 70 (2019) 547–562.
[11] R. Chongtay, M. Last, B. Berendt, Responsive news summarization for ubiquitous consumption on multiple mobile devices, 23rd International Conference on Intelligent User Interfaces (2018) 433–437.
[12] C. Comito, D. Talia, Energy consumption of data mining algorithms on mobile phones: evaluation and prediction, Perv. Mob. Comput. Elsevier 42 (2017) 248–264.
[13] L. Corral, A.B. Georgiev, A. Sillitti, G. Succi, A study of energy-aware implementation techniques: REDISTRIBUTION of computational jobs in mobile apps, Sustain. Comput. Informatics Syst. 7 (2015) 11–23.
[14] D. Das, A.F. Martins, A survey on automatic text summarization, Literature Survey for the Language and Statistics II Course at CMU 4 (2007) 192–195.
[15] D.D. Deb, S. Bose, S. Bandyopadhyay, Coordinating disaster relief operations using smart phone/pda based peer-to-peer communication, Int. J. Wirel. Mob. Netw. 4 (2012) 27.
[16] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding, 2018. arXiv:1810.04805.
[17] J. Dolezal, Z. Becvar, Methodology and tool for energy consumption modeling of mobile devices, 2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW) (2014) 34–39.
[18] Y. Dong, A Survey on Neural Network-Based Summarization Methods, 2018. arXiv: 1804.04589.
[19] W.S. El-Kassas, C.R. Salama, A.A. Rafea, H.K. Mohamed, Automatic text summarization: a comprehensive survey, Expert Syst. Appl. (2020) 113679.
[20] G. Erkan, D.R. Radev, Lexrank: Graph-based lexical centrality as salience in text summarization, J. Artif. Intell. Res. 22 (2004) 457–479.
[21] R. Ferreira, L. de Souza Cabral, R.D. Lins, G.P. e Silva, F. Freitas, G.D. Cavalcanti, R. Lima, S.J. Simske, L. Favaro, Assessing sentence scoring techniques for extractive text summarization, Expert Syst. Appl. 40 (2013) 5755–5764.
[22] O.M. Foong, S.P. Yong, F.A. Jaid, Text summarization using latent semantic analysis model in mobile android platform, 2015 9th Asia Modelling Symposium (AMS) (2015) 35–39.
[23] O.M. Foong, S.P. Yong, A.L. Lee, Text summarization in android mobile devices, Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013) (2014) 571–578.
[24] M. Gambhir, V. Gupta, Recent automatic text summarization techniques: a survey, Artif. Intell. Rev. 47 (2017) 1–66.
[25] K. Ganesan, Rouge 2. 0: Updated and Improved Measures for Evaluation of Summarization Tasks, 2015.
[26] X. Gao, D. Liu, D. Liu, H. Wang, A. Stavrou, E-android: a new energy profiling tool for smartphones, 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS) (2017) 492–502.

[27] A. Gember, A. Anand, A. Akella, A comparative study of handheld and non-handheld traffic in campus wi-fi networks, International Conference on Passive and Active Network Measurement (2011) 173–183.

[28] F.B. Goularte, S.M. Nassar, R. Fileto, H. Saggion, A text summarization method based on fuzzy rules and applicable to automated assessment, Expert Syst. Appl. 115 (2019) 264–275.

[29] P.K. Gupta, P.K. Muhuri, Per-c based green computing model for handheld devices: an application of single person fou, 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (2016) 1–6.

[30] V. Gupta, G.S. Lehal, A survey of text summarization extractive techniques, J. Emerg. Technol. Web Intell. 2 (2010) 258–268.

[31] S. Hao, D. Li, W.G. Halfond, R. Govindan, Estimating mobile application energy consumption using program analysis, Proceedings of the 35th International CONFERENCE on Software Engineering (2013).

[32] A. Hatipoglu, S.İ. Omurca, A turkish wikipedia text summarization system for mobile devices, Int. J. Inform. Technol. Comput. Sci. 1 (2016) 1–10.

[33] G.B.D.L. Inference, A Performance and Power Analysis, Whitepaper, 2015. November.

[34] S. Kim, L. Yeganova, W.J. Wilbur, Summarizing topical contents from PubMed documents using a thematic analysis, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (2015).

[35] S. Kumar, X. Hu, H. Liu, A behavior analytics approach to identifying tweets from crisis regions, Proceedings of the 25th ACM conference on Hypertext and social media (2014).

[36] L.A. Leiva, Responsive text summarization, Inform. Process. Lett. 130 (2018) 52–57.

[37] X. Liu, Y. Li, F. Wei, M. Zhou, Graph-based multi-tweet summarization using social signals, Proc. COLING 2012 (2012) 1699–1714.

[38] Y. Liu, M. Lapata, Text Summarization With Pretrained Encoders, 2019. arXiv: 1908.08345.

[39] Z. Lu, G. Cao, T. La Porta, Teamphone: networking smartphones for disaster recovery, IEEE Trans. Mob. Comput. 16 (2017) 3554–3567.

[40] H.P. Luhn, The automatic creation of literature abstracts, IBM J. Res. Dev. 2 (1958) 159–165.

[41] C. Mallick, A.K. Das, M. Dutta, A.K. Das, A. Sarkar, Graph-based text summarization using modified textrank, Soft Computing in Data Analytics (2019) 137–146.

[42] A. Meneses-Viveros, E. Hernandez-Rubio, S. Mendoza, J. Rodriguez, A.B. M. Quintos, Energy saving strategies in the design of mobile device applications, Sustain. Comput. Informatics Syst. 19 (2018) 86–95.

[43] J. Messias, J.P. Diniz, E. Soares, M. Ferreira, M. Araujo, L. Bastos, M. Miranda, F. Benevenuto, Towards sentiment analysis for mobile devices, Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (2016).

[44] C. Meurisch, S. Wullkotte, S. Niemczyk, F. Kohnh“auser, M. M” uhlhäuser, et al., Nicer911: ad-hoc communication and emergency services using networking smartphones and wireless home routers, Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing (2017) 33.

[45] J.C. Miller, G. Rae, F. Schaefer, L.A. Ward, T. LoFaro, A. Farahat, Modifications of kleinberg's hits algorithm using matrix exponentiation and web log records, Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2001).

[46] R. Mittal, A. Kansal, R. Chandra, Empowering developers to estimate app energy consumption, Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (2012).

[47] I.F. Moawad, M. Aref, Semantic graph reduction approach for abstractive text summarization, Proceedings of the 7th International IEEE Conference on Computer Engineering & Systems (ICCES) (2012).

[48] V. Myasnikov, S. Sartasov, I. Slesarev, P. Gessen, Energy consumption measurement frameworks for android os: a systematic literature review, Fifth Conference on Software Engineering and Information Management (SEIM-2020) (Full Papers) (2020) 18.

[49] A. Nenkova, L. Vanderwende, K. McKeown, A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization, Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2006).

[50] J. Otterbacher, D. Radev, O. Kareem, News to go: hierarchical text summarization for mobile devices, Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2006).

[51] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford InfoLab, 1999.

[52] S. Park, B. Cha, K. Chung, J. Kim, Mobile iot device summarizer using p2p web search engine and inherent characteristic of contents, Peer-to-Peer Network. Appl. 13 (2020) 684–693.

[53] Z. Ren, S. Liang, E. Meij, M. de Rijke, Personalized time-aware tweets summarization, Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (2013).

[54] S. Robinson, Cellphone energy gap: desperately seeking solutions, Strat. Anal. (2009).

[55] K. Rudra, et al., Extracting situational information from microblogs during disaster events: a classification-summarization approach, Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (2015).

[56] H. Sahar, A.A. Bangash, M.O. Beg, Towards energy aware object-oriented development of android applications, Sustain. Comput. Informatics Syst. 21 (2019) 28–46.

[57] M. Solutions, Inc., Monsoon Power Monitor, 2017.

[58] S. Song, H. Huang, T. Ruan, Abstractive text summarization using lstm-cnn based deep learning, Multimedia Tools Appl. 78 (2019) 857–875.

[59] J. Steinberger, et al., Using latent semantic analysis in text summarization and summary evaluation, Proc. ISIM 4 (2004) 93–100.

[60] K. Tao, et al., Groundhog day: near-duplicate detection on twitter, Proceedings of the Conference on World Wide Web (WWW) (2013).

[61] L. Vanderwende, H. Suzuki, C. Brockett, A. Nenkova, Beyond sumbasic: task-focused summarization with sentence simplification and lexical expansion, Inform. Process. Manag. Elsevier 43 (2007) 1606–1618.

[62] Z. Wang, L. Shou, K. Chen, G. Chen, S. Mehrotra, On summarization and timeline generation for evolutionary tweet streams, IEEE Trans. Knowl. Data Eng. 27 (2015) 1301–1315.

[63] Z. Wei, W. Gao, Gibberish, assistant, or master? Using tweets linking to news for extractive single-document summarization, Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (2015).

[64] Z. Wei, Y. Liu, C. Li, W. Gao, Using tweets to help sentence compression for news highlights generation, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (2015) 50–56.

[65] C.C. Yang, F.L. Wang, Fractal summarization for mobile devices to access large documents on the web, Proceedings of the 12th International Conference on World Wide Web (2003).

[66] H. Zhang, J. Xu, J. Wang, Pretraining-Based Natural Language Generation for Text Summarization, 2019. arXiv:1902.09243.