Software Impacts 17 (2023) 100543

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

ShortMail: An email summarizer system

Mahira Kirmani [a], Gagandeep Kaur [a], Mudasir Mohd [b,*]

[a] *University Institute of Computing, Chandigarh University, NH-05 Chandigarh-Ludhiana, India*
[b] *South Campus University of Kashmir, India*

ARTICLE INFO

Keywords:
Text summarization
Extractive text summarization
Semantic models
BERT

ABSTRACT

On average, we receive approximately more than 40 emails per day and spend around 2 min reading each email, i.e., 80 min a day are spent by the user reading emails. This time can be saved if the user is presented with a summary of emails. Thus we propose *ShortMail*, an email summarizing system. *ShortMail* produces summaries of emails composed in English-language. Furthermore, it allows users to perform follow-up actions on these emails and is thus a one-stop solution for all emailing functionalities. *ShortMail* uses stat-of-art Semantic models and deep-learning technologies to summarize emails and is a potential solution to information overload.

Code metadata

Current code version	*V1*
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2023-240
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/7372043/tree/v1
Legal Code License	*GNU General Public License v3.0*
Code versioning system used	*git*
Software code languages, tools, and services used	*Python and Flask*
Compilation requirements, operating environments & dependencies	*Python*3.6 >=
If available Link to developer documentation/manual	https://github.com/mudasirmohd/Shortmails.git
Support email for questions	mudasir.mohammad@kashmiruniversity.ac.in

1. Introduction

Text summarization is a challenging problem in Natural Language Processing (NLP) [1], which involves condensing the content of textual documents without losing their overall meaning and information content. Text summarizers take as input a lengthy textual document that analyzes and processes them to produce a gist for immediate consumption, with the goal of retaining the maximum information content of the original document. The resultant summaries are hence easier to comprehend and understand. Text summarizer is the potential solution to the information overload problem [2].

Emails, short for electronic mail, are messages or letters sent and received via the internet or a computer network. They are digital messages that can include text, images, attachments, and links, and are sent from one person to another or to multiple recipients simultaneously. Emails are used for various purposes: communication, marketing, advertising, and sharing information or documents. They are often used for business purposes and have become essential to modern communication and work processes. According to the report by Radicati Group, Inc[1] the total number of emails sent all over the world per day are 347.3 billion and there are 7.73 billion email users currently. Both are growing at the rate of 4.3% and 3%, respectively.

With this volume of information received through emails, it becomes overwhelming for the receiver to read every email in detail. Thus we propose *ShortMail*, an email summarizer tool for summarizing emails. It summarizes users' emails for quick consumption. Our *ShortMail* helps the receiver by providing information clearly and concisely.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: mahirakirmani68@gmail.com (M. Kirmani), gagandeepkaurlogani@gmail.com (G. Kaur), mudasir.mohammad@kashmiruniversity.ac.in (M. Mohd).

[1] https://www.radicati.com/wp/wp-content/uploads/2020/12/Email-Statistics-Report-2021-2025-Executive-Summary.pdf

https://doi.org/10.1016/j.simpa.2023.100543
Received 11 June 2023; Received in revised form 25 June 2023; Accepted 2 July 2023

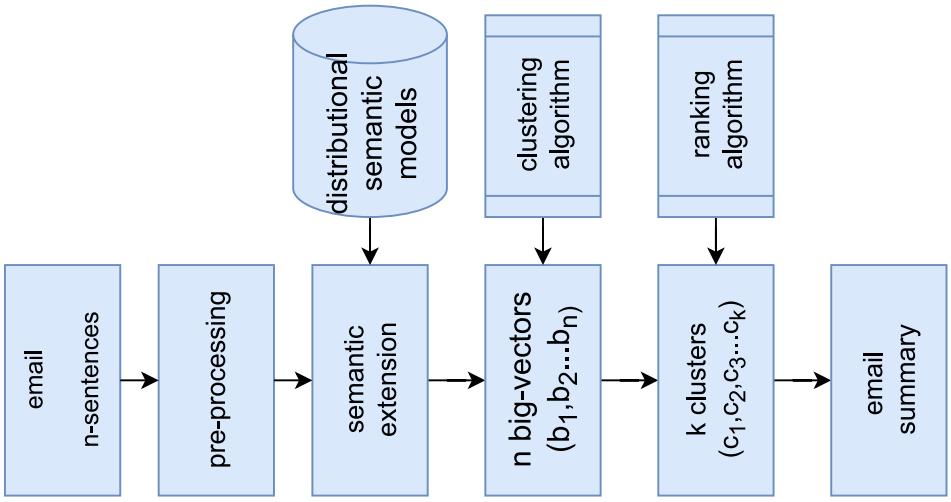

Fig. 1. Overall functioning of TSE.

Recipients can quickly decide whether a particular email needs immediate action and decide the action based on the summaries. Clear and concise summaries help the recipients understand the main message and initiate necessary actions or follow-ups. Overall our *ShortMail* saves time and improves communication efficiency for both sender and the receiver.

In this paper, we present the development and evaluation of *ShortMail*, highlighting its effectiveness in summarizing emails. We discuss the underlying methodology and techniques employed by *ShortMail* to achieve accurate and informative summaries. Through this research, we aim to alleviate the burden of email overload and streamline email management processes, ultimately enhancing productivity in today's fast-paced digital environment.

2. Functionalities

ShortMail is an end-to-end summarization system. Its backend has the summarizing engine (TSE), and the front end is a web-based system. The user has to set up the mailbox by forwarding the emails to *ShortMail*, where TSE uses state-of-art NLP and deep learning algorithms to summarize the email. The end results are then displayed to the user using the *ShortMail's* web interface.

Thus, the software has two essential components TSE and the web interface. TSE is the heart of the system and employs our novel ranking algorithm used in [2] with some modifications to work with the emails. The overall structure of our TSE is explained in Fig. 1. TSE takes input as emails forwarded from the user's inbox and processes each email to produce summaries. TSE uses the following steps to summarize emails

2.1. Pre-processing

Before any email is processed for summarization, we clean the emails so as to remove unnecessary parts from emails that are not required for summarization. Pre-processing is also done to remove email inconsistencies and make data uniform.

- **Remove irrelevant content-** Emails received from the user's mailbox comprised of different parts, including subject, body, signature and attachments. We remove the subject, signature and attachments to obtain a summary of the email.
- **Remove thread emails-** Many emails are continuous emails which arrive in the mailbox as threads. We remove these threads and keep only the latest email.
- **Remove punctuations -** Numbers, currency signs and punctuations convey no meaningful information content and are thus removed from the input text.
- **Tokenisation-** After cleaning, every sentence in the email is tokenized into words. *Shortmail* uses Stanford Core NLP [3] for tokenization.
- **Lowercase-** All the text is converted to lowercase for efficient processing.
- **Lemmatization-** Lemmatization converts text to its base form without losing meaning. We lemmatize the entire text to its base form to make data uniform. We use the Stanford core NLP package to perform lemmatization.

2.2. Semantic extension

Semantics form an inherent and crucial feature of the input document but are overlooked by the existing summarizers in the literature. Hence the existing summarizers are overlooking this aspect of the text. These summarizers assume that only statistical features are central to the summarization and thus miss out on semantics. *ShortMail* uses text semantics as a feature to obtain summaries. It uses the distributional semantic model to obtain the semantics of text. We employ Google's Bidirectional Encoder Representations from Transformers (BERT). These models do not require any lexical and linguistic analysis. Furthermore, these models are independent of external information to obtain semantics. BERT is used to obtain semantics in various fields of NLP like Question Answering systems [4,5], Neural Machine translation [6], and Relation Extraction [7].

2.3. Big-vector generation

Big-vector generation is our novel algorithm, generating big-vectors from the vectors obtained using the distributional bio-semantic models. For all the words of the input text, we feed these words to the different bio-semantic models to achieve the concatenated vectors of the sentence. The concatenated vectors form a rich bag of words like a single vector for the sentence.

Let $\delta(w)$ is a function for retrieving a top list of 'm' words from a semantic model. The function is given as $w' = \delta(w) = w'_1 \oplus w'_2 \oplus, \ldots, \oplus w'_m$. For a sentence with $W = \{w_1, w_2, w_3, \ldots., w_k\}$ as the sequence of k tokenized words, a big-vector BGV is populated by concatenating respective top m similar words for each word i.e $BGV = \{\delta(w_1) \oplus \delta(w_2) \oplus, \ldots, \oplus \delta(w_k)\}$.

This rich semantic vector is then fed to the clustering algorithm to obtain different clusters representing all the semantic information for that cluster. The algorithm for Big-vector generation is given in the algorithm 1. Once we obtain the rich big-vectors, these vectors are

Algorithm 1 Algorithm for Big-vector generation

Let D be input email
s_i is the sentences of email D

for all $s_i \in$ D **do**
  $W \leftarrow Tokenization(s_i)$
  where $W = \{w_1, w_2, w_3, ..., w_n\}$

  for all $w_i \in W$ **do**
    $V_i \leftarrow BERT(W_i)$
  end for
  $BV = V_1 \oplus V_2 \oplus \cdots \oplus V_{|W|}$
  $\oplus$ is concatenation
end for

clustered into different clusters using the K-means clustering algorithm. The objective here is to divide the sample space of these rich big-vectors into different semantic clusters so that each cluster will contain semantically similar sentences in a single semantic group. We use K-means [8] clustering algorithm to form the semantically rich big-vectors clusters. The input for clustering is the big-vectors, and we obtain semantic clusters of sentences as output. We then use our novel ranking algorithm to retrieve the top n sentences from each cluster.

2.4. Ranking algorithm

Our novel ranking algorithm aims to assign sentence ranks to the sentences of emails according to different surface-level features. The features that we use here are sentence position, Frequency (TF-IDF), proper nouns, and cosine similarity. The total score of the sentence is then calculated by summing the individual normalized scores of each sentence.

Web interface allows the user to interact with the system. It is built using the *Flask*. It notifies the user whenever a new email arrives. Users can check the email summary and then take actions like deleting it, checking the original email or replying to it.

3. Impacts

The impact of *ShortMail* extends beyond its function as a tool for generating real-time email summaries. By forwarding emails from the user's mailbox to our system, *ShortMail* analyzes and processes the content, generating concise summaries. This approach significantly reduces the burden of email overload, allowing users to manage their emails more efficiently and save valuable time. The ability to quickly grasp the essence of an email through a summary empowers users to prioritize their actions and respond promptly to important messages.

Furthermore, *ShortMail* offers various practical applications that showcase its versatility and usefulness. In addition to integrating with individual users' mailboxes, *ShortMail* has been successfully implemented in several software projects across different domains. For example, BioSum [2,9] leverages the summarization capabilities of *ShortMail* to condense and extract key information from complex biomedical papers. This application demonstrates how *ShortMail* contributes to streamlining literature review processes and aids researchers in quickly identifying relevant insights.

Moreover, the Semantic Summarizer[2] utilizes the summarization functionality of *ShortMail* as a core component. This general-purpose summarizer for English language text highlights the adaptability of *ShortMail* across various domains and its contribution to enabling efficient information extraction from diverse textual sources.

Additionally, the industrial adoption of *ShortMail* in the software application[3] further demonstrates its real-world impact. In the final industrial phase, users access email summaries directly on their mobile devices using a dedicated mobile app. This integration enhances user productivity and underscores the practicality and applicability of *ShortMail* in professional settings.

By showcasing these real-world applications and their tangible benefits, it becomes evident that *ShortMail* goes beyond being a mere software package. It impacts various domains, from individual email management to research literature review processes and industrial productivity. The practical implications of *ShortMail* are underscored by its integration into existing software projects and its ability to simplify and streamline complex information processing tasks.

4. Conclusion and future improvements

ShortMail can be used by the users to receive summaries for all emails delivered to them. These emails are forwarded from their mailboxes into the *ShortMail*, where they are summarized and allow users to take follow-up actions based on the email summaries. *ShortMail* aims to reduce the information overload of the user. It works on English language email communication. It provides a one-stop solution for all email functioning. Users can view, delete and reply to emails using it.

In future versions of the software, we aim to make *ShortMail* multilingual. Our work in creating semantic models for low-resource languages will lead us to develop semantic models for these languages. Thus, we can extend *ShortMail* to work on multi-lingual emails.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Mohd, S. Javeed, Nowsheena, M.A. Wani, H.A. Khanday, Sentiment analysis using lexico-semantic features, J. Inf. Sci. (2022) 01655515221124016.
[2] M. Mohd, R. Jan, M. Shah, Text document summarization using word embedding, Expert Syst. Appl. 143 (2020) 112958.
[3] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Association for Computational Linguistics (ACL) System Demonstrations, 2014, pp. 55–60, URL http://www.aclweb.org/anthology/P/P14/P14-5010.
[4] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (4) (2020) 1234–1240.
[5] M. Mohd, R. Hashmy, Question classification using a knowledge-based semantic kernel, in: M. Pant, K. Ray, T.K. Sharma, S. Rawat, A. Bandyopadhyay (Eds.), Soft Computing: Theories and Applications, Springer Singapore, Singapore, 2018, pp. 599–606.
[6] D. Kim, J. Lee, C.H. So, H. Jeon, M. Jeong, Y. Choi, W. Yoon, M. Sung, J. Kang, A neural named entity recognition and multi-type normalization tool for biomedical text mining, IEEE Access 7 (2019) 73729–73740.
[7] C. Lin, T. Miller, D. Dligach, S. Bethard, G. Savova, A BERT-based universal model for both within-and cross-sentence clinical temporal relation extraction, in: Proceedings of the 2nd Clinical Natural Language Processing Workshop, 2019, pp. 65–71.
[8] J.A. Hartigan, M.A. Wong, Algorithm AS 136: A k-means clustering algorithm, J. R. Stat. Soc. C (Appl. Stat.) 28 (1) (1979) 100–108.
[9] I.K. Bhat, M. Mohd, R. Hashmy, SumItUp: A hybrid single-document text summarizer, in: M. Pant, K. Ray, T.K. Sharma, S. Rawat, A. Bandyopadhyay (Eds.), Soft Computing: Theories and Applications, Springer Singapore, Singapore, 2018, pp. 619–634.

[2] https://gitlab.com/mudasir-mohd/semantic-summarizer.git

[3] https://quickwordz.com