Pattern Recognition 144 (2023) 109839

Contents lists available at ScienceDirect

Pattern Recognition

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

Unsupervised update summarization of news events

Florian Carichon [a,*], Florent Fettu [b], Gilles Caporossi [a]

[a] HEC Montreal, 3000 Chem. de la Cote-Sainte-Catherine, Montreal, H3T 2A7, Quebec, Canada
[b] Federation des Caisses Desjardins du Quebec, 175 Boulevard Rene-Levesque O, Montreal, H2X 3Y2, Quebec, Canada

ARTICLE INFO

MSC:
62M45
68T50

Keywords:
Natural language processing
Neural network
Automatic document summarization
Unsupervised approach
Update sentence compression
Information novelty

ABSTRACT

A long-running event represents a continuous stream of information on a given topic, such as natural disasters, stock market updates, or even ongoing customer relationship. These news stories include hundreds of individual, time-dependent texts. Simultaneously, new technologies have profoundly transformed the way we consume information. The need to obtain quick, relevant, and digest updates continuously has become a crucial issue and creates new challenges for the task of automatic document summarization. To that end, we introduce an innovative unsupervised method based on two competing sequence-to-sequence models to produce short updated summaries. The proposed architecture relies on several parameters to balance the outputs from the two autoencoders. This relation enables the overall model to correlate generated summaries with relevant information coming from both current and previous news iterations. Depending on the model configuration, we are then able to control the novelty or the consistency of terms included in generated summaries. We evaluate our method on a modified version of the TREC 2013, 2014, and 2015 datasets to track continuous events from a single source. We not only achieve state-of-the-art performance similar to other more complex unsupervised sentence compression approaches, but also influence the information included in the model in the summaries.

1. Introduction

Automatic text summarization is the process of distilling information contained in one or more documents to produce a reduced version that meets the need of a particular task or user. The most frequently used data source in document summarization is news events. Because news events are inherently time-sensitive [1], update summarization was one of the first tasks emerging in the 2008 Document Understanding Conferences[1] (DUC) and subsequently taken up by the Text Analysis Conferences[2] (TAC) and the Text REtrieval Conference[3] (TREC) in the temporal and real time summarization tracks. Of course, news information is now consumed via new media. To wit, people are increasingly turning to blogs, web journals and Twitter for their news to, for example, keep up to date on developing events such as natural disasters [2]. Consumers, especially young people, are also increasingly likely to follow the news live through their smartphones. Specifically, more than 55% of smartphone users receive notifications on their phones alerting them to breaking news or major events. More crucially, half of these users will consult the full article after reading a notification.[4] Notification quality is then crucial because it provides some basic information to users who opt to not read the full article, while simultaneously increasing the likelihood that users will click though. In this regard, the notifications act as an efficient real-time summary of a given event that is further fleshed out with each subsequent short headline. The objective of update summarization is to produce outputs that include both relevant and new content that factors in some background knowledge, represented by previously generated material [3]. For events lasting for longer periods, that background information is iteratively enriched with new data [4,5]. Thus, relevance and novelty are reassessed with every update to determine what information to include in the summary [6]. The most efficient methods then consist in simultaneously scoring and selecting sentences depending on relevance and novelty to meet the user's need for information while guaranteeing the independence and quality of the sentences incorporated in the results [7,8]. However, most of these methods ease the characterization salient and novel content through the analysis of redundant information in multiple sources.

In this paper, we introduce a new method to conduct update summarization for short single documents. Our approach has several advantages over existing methods. Since the documents and summaries

* Corresponding author.
E-mail addresses: florian.carichon@hec.ca (F. Carichon), florent.fettu@desjardins.com (F. Fettu), gilles.caporossi@hec.ca (G. Caporossi).

[1] https://duc.nist.gov/
[2] https://tac.nist.gov/about/index.html
[3] https://trec.nist.gov/
[4] http://pewrsr.ch/2ccvrnC

https://doi.org/10.1016/j.patcog.2023.109839
Received 1 July 2022; Received in revised form 3 July 2023; Accepted 22 July 2023
Available online 31 July 2023

Abbreviations

LM	Language Model	
CM	Constraint Model	
Seq2Seq models	Sequence to Sequence models	
TFIDF	Term-Frequency/Inverse Document Frequency	

are single short texts, we cannot rely on classic techniques based on redundancy to identify salient content. Indeed, these approaches are mobilized at the news story level and thus require local redundancy of information within this news to estimate the relevance of the terms. They then define novelty as any material that is not redundant [6,9]. We also depend on local information to measure relevance and novelty, but we combine and constrain it with global metrics such as the Term-Frequency/Inverse Document Frequency (TFIDF). Crucially, this reduces the need for local redundancy and therefore makes it possible to handle single short documents. Moreover, these methods are generally designed for multiple documents and are thus extractive, which prevents them from producing coherent outputs in this context. Neural network systems applied to tasks such as update or progressive update summarization have proven to efficiently create abstractive summaries of single documents [10]. These approaches rely on the model capacity to identify relevant and new information in supervised fashion through the huge quantity of labeled samples. However, the example in 1 illustrates that this data is acutely noise, which will prevents supervised models from functioning as intended. Furthermore, in many real-world scenarios, the diversity of topics and genres of the data compromises access to the labels required to train those systems. Therefore, we assume it is better to increase the portability of our model by favoring unsupervised techniques. Recent neural network models such as autoencoders have demonstrated their efficiency to capture important content and generate unsupervised abstractive summaries [11]. The main matter remains to find means to constrain the content and the length of the produced summary. Unsupervised approaches have been adapted to process short inputs by combining information selection techniques with semi-abstractive text generation [12,13]. Despite these very promising results, none of these methods account for past summaries for a given event. However, the reference examples employed in the Fig. 1 point up the importance of considering this temporality to ensure the summary relevance and cohesion for the whole news story. For this reason, and as we can see, our approach compares the new information with the content it has previously generated. Moreover, we decide to use the summary as an input instead of the original text in light of extant research exposing that only half of users consult the full article after receiving a notification.

This paper builds on the previous work done on the TREC temporal track of datasets. Once again, most approaches to this task generate summaries through the analysis of redundant information in multiple sources [5]. Therefore, to perform our task of update summarization on single documents, we combine the 2013, 2014, and 2015 TREC temporal summarization tracks and modify them to follow sources individually. The left part of Fig. 1 provides a sample news story (a train crash in Argentina) that could be extracted in this manner. It is also a case in point as to why relevant and updated live notifications are crucial for developing news stories. The news comes from a single source, and information is clearly reused through each iteration. For instance, the term “railway” is recycled from the original information in the first update summary to ensure consistency. We can also notice the application of novel terms such as “Argentina’s transportation secretary” assumes the reader viewed the previous summary about the “Transport Secretary Juan Pablo Schiavi”. The text on the right of Fig. 1 supports our intuition about the importance of allowing the model to control the novelty and/or consistency of information through the iterations to include content that would not be available otherwise. In addition, one significant aspect that we can observe in the example is the discrepancy of data due to our modification for single update summarization. For instance, one reference text states the number of “676+ injuries”; information which cannot be retrieved from either the source text or the previous iteration. This phenomenon makes it difficult to recognize the origin of the content, and why the material has been included in the final summary.

In response to the above challenges, we present an unsupervised autoencoder model where the summary generation is influenced by the novelty and coherence of the information previously provided. More specifically, the model relies on a Sequence-to-Sequence (Seq2Seq) architecture that simultaneously learns a language model (LM) and follows an information constraint model (CM) composed by a reduced length version of text with only the best Term-Frequency/Inverse Document Frequency (TFIDF) scoring words. The LM receives an encoded Recurrent Neural Network (RNN) representation of the concatenation of the previously generated summary and the current text. The objective function enforces learning to reconstruct the information from both sources. Then, the same RNN encoder produces a representation for the reduced version of the current text that contains its informative content. Terms are selected based on their TFIDF values and weighted by their occurrence in the previous summary. The objective function is to recreate this reduced text. This second term serves both to enforce the model’s length constraint and to provide guidance to select information to include in the final summary. The inclusion in both parts of the prior content makes it possible for the model to decide what material from the previous steps will be retained. Finally, in the generation step, the model must choose between following the LM or the informative content.

As such, this paper makes the following contributions:

- We introduce the new task of update summarization of short single documents. This task can also be used to achieve update sentence compression if the document is composed of a single sentence.
- We propose a modification of the TREC temporal summarization dataset to perform and evaluate this task.
- We present a novel unsupervised semi-extractive summarization approach composed by two competing auto-encoding models to generate summaries of short documents. This combined structure selects the most likely terms from either a language model that reconstruct grammatical and coherent texts or a length constraining model that only enforces the selection of salient words.
- To perform update summarization, we have also introduced a new hyperparameter modifying the behavior of both models. First, the parameter is employed in the learning objective of the language model to reconstruct the current text while considering the previous generated summaries. In the information constraint model, we use this parameter to weight the score of the relevant terms of the current text by their occurrence in the precedent iterations.

We assess the performance of our model using the standard ROUGE [14] and SUPERT [15] document evaluation metrics. Based on these metrics, our model’s performance is either equivalent or better than more complex unsupervised baselines as well as certain supervised baselines. We also conducted human evaluations to judge the relevance and the coherence of the summaries. The results show that our model encompasses more salient information and, despite being less grammatically coherent than human references, it nonetheless produces understandable and reasonable texts. Finally, using both manual and automatic methods, we estimates the novelty and consistency of the summaries produced by different approaches, demonstrating that our model can be used to control information included in the output.

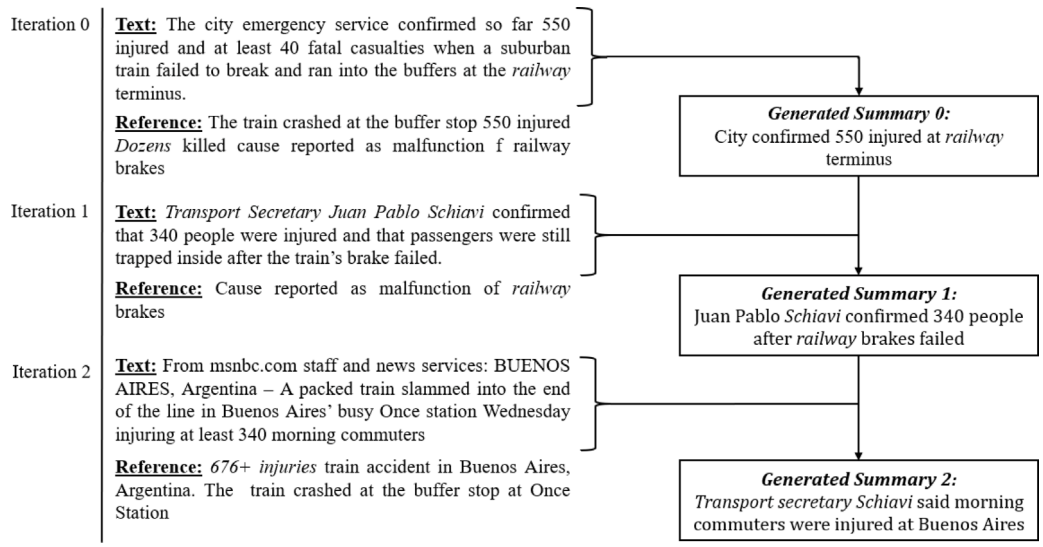

Fig. 1. The left side introduce a news event concerning a train crash in Argentina. The event is updated 3 times. The *italicized text* in the left-hand column highlights which terms can be reused through iterations. It also displays discrepancies between sources and reference summaries. The text on the right-hand column demonstrates how our model uses generated content to produce improved iterative summaries.

2. Related work

This paper draws on three topics of automatic text summarization, described below:

2.1. Unsupervised summarization

Classical approaches of unsupervised summarization have prioritized extractive methods that optimize the selection of salient, representative, and diverse sentences form one or multiple texts. More recent approaches relying on sub-modular functions and hand-crafted features [16], or the use of pre-trained models and a determinantal point process technique to account for redundancy [17] have demonstrated the interest of creating precise optimization functions to meet these criteria in order to achieve strong task performance. The emergence of deep learning techniques led to the generalization of abstractive approaches, as these create entirely new texts as summaries. Unsupervised models need a limit to produce a reduced version of the original inputs. For multiple documents, the constraint is implicit. Indeed, the objective is to learn a representation of the set of documents which will be decoded as an average of the input [11]. For single document summarization, the constraint must be explicit and applied to the size of the original or produced text. This method is also applicable for new semi-extractive approaches where the model learns to generate a compressed version of the input sentence. The objective is then to set a constraint such that the model drops non-informative words from the original sentence to form the summary. The first approach suggests using a denoising autoencoder where authors add additional grammatical non-informative words to the input and the model learns to omit them afterwards to create the compression [12]. Since compression coerces the system to remove content, some methods modify the CM to help the model apprehend the salient information. The model is then forced to respect new constraints depending on the main topics [13] of the documents, improbable informative words [18], or the mutual information between the original sentence and the compression [19]. In this article we also introduce a semi-extractive approach with constraints to facilitate information capture for the summary. It differs from previous work because we apply a pretrained TFIDF model, which shows very good performance for news stories. Moreover, this simple and more explicit model allows us to account for previously generated information by updating the original TFIDF score with the previous word occurrences.

2.2. Novelty and consistency

The novelty principle is a fundamental concept for many NLP tasks such as information retrieval, Q&A systems, recommender systems, or document summarization. The goal is to offer a sufficient variety of information to users so that at least one item meets their expectations [7]. To date, the most influential method in the field remains the Maximum Margin Relevance (MMR) [8]. It introduces a ranking algorithm that selects relevant sentences and penalizes them based on their similarity to the ones already included in the summary. The MMR approach operates at the sentence selection step, but novelty can be estimated through the three steps of the summarization process: scoring, selection, and summary generation [4]. Recently, several approaches based on the Seq2Seq model have tried to implement novelty systems in the summarization process at the scoring level [20]. The model that most closely resembles ours implements the Maximum Mutual Information (MMI) in the objective function of the model to generate diverse responses in a discussion [10]. The authors argue in favor of integrating the MMI in the model objective function since it can capture inter-sentence relations. Although novelty is crucial to ensure the relevance of information, the consistency of consecutive content is also of utmost importance to evaluate what to include in the summary. Multiple applications in video summarization have highlighted the fundamental role of similar contextual frames for summary generation [21,22]. Another approach further demonstrated the importance of diversifying attention paid to the context to improve salience estimation [23]. Finally, a recent approach demonstrated that using unsupervised learning on temporal and contextual data combined with reinforce learning techniques is useful for iterative or dynamic outputs [24]. In the case of textual data, the consistency of information-characterized, for instance, by the presence of related lexical items and smooth semantic transitions—is also key to ensuring the users' comprehension of the summary [25]. Recent methods have focused on guaranteeing sentence reordering in the summary based on information found in previous sentences to ensure these semantic associations [26]. A summary emphasizing consistency will then highlight the aboutness and indicativity of the summary [25] while the novelty will promote the informativeness of the text [1]. Our approach accounts for these potential relations by linking the current text and the previous model output. More specifically, we added the notion of redundancy in the scoring of informative words to preserve. The constraint input is composed only of the highest TFIDF scores, which is then weighted by the

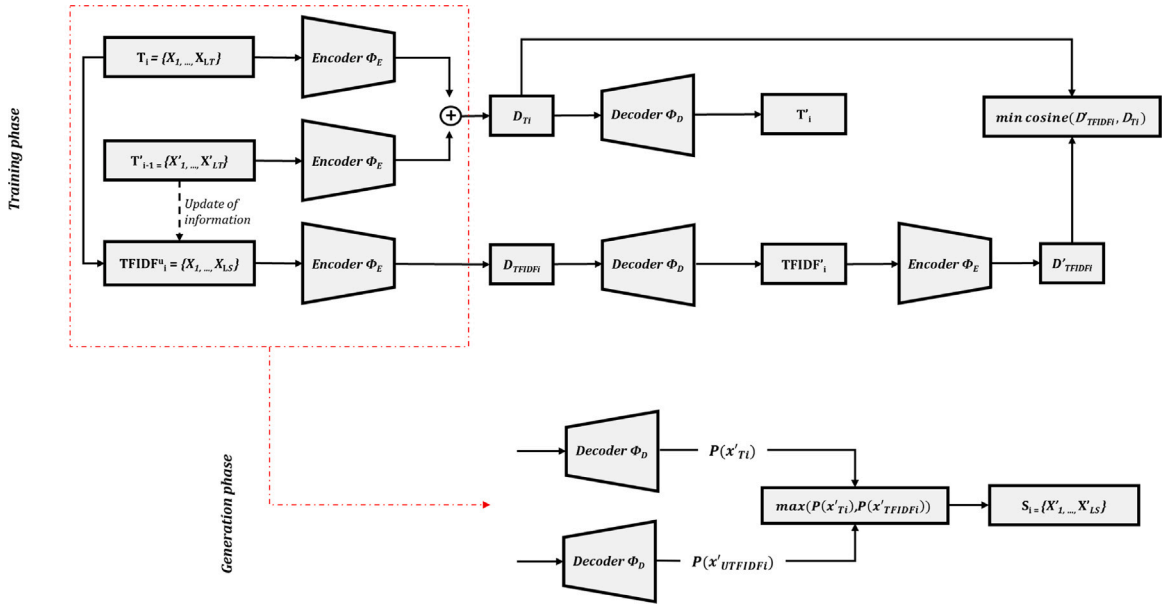

Fig. 2. Autoencoder architecture for update summarization. At each text iteration T_i, we use as additional input the previously generated summary T'_{i-1} to produce the current representations D_{Ti} which is used to train the language model. We also use T'_{i-1} to generate $TFIDF''_i$, the vector composed of the most relevant updated terms. This representation is encoded into D_{TFIDFi} to train the information constraint model. At the generation stage, we select the highest probability between T'_i or $TFIDF'_i$ to form the final summary.

frequency obtained in the previous summary and by a MMR method for reconstructing the documents. The model is then able to select either the word in the language model that foster appropriate for new inter-sentence relations/associations or words that maintain consistent update information.

2.3. Update summarization

The goal of the update summarization task is to produce a summary that focuses on new relevant facts for users. The scope of the update can include single updates or the temporal tracking of a continuous stream of new incoming documents that are slated for summarization. It may also be a progressive creation of new material or the dynamic modification of the output at specific timed intervals [4]. As the system deals with an unfolding event, the information salience, which is often based on content redundancy, become too complex to evaluate over time [6]. To facilitate this process, the event is considered as a set of documents, which opens the door to the application of classical multi-document summarization (MDS) methods. The major challenge is to come up with new material compared to previous iterations. In the first attempt, the authors implemented a ranking algorithm that multiplied the probabilities of the relevance and novelty of terms to score and subsequently include them in each update [3]. The same principle was then used for the MMR approach and has been proved efficient for identifying true and relevant information in the context of summarization for improving fake news detection [27]. Other techniques propose to dynamically weight the novelty and relevance thresholds in ranking systems using the quantity of information present in each period [5]. Finally, instead of using similarity and redundancy, other authors have handled the task by diversifying the topics covered in the updated output by including information from different document clusters or topics [9]. However, all of these approaches overlap insofar as they introduce extractive multi-document summaries of the original content. Neural networks models such as Seq2Seq architectures make it possible to generate a new piece of text conditioned by temporal information. These abstractive models can create compressions applicable for both single and multiple documents summaries. Particularly, this method has been applied to single document summarization where the conditioning information is taken from an external source such as Wikipedia for knowledge transfer [28]. It has also been employed to manage the salience and the consistency of updated content for real-time streams of multiple tweets [29]. Our approach is also applied to single document update summarization; however, one key difference is that our approach is unsupervised and therefore more portable to different fields. Moreover, the use of frequency and TFIDF to estimate term importance make it possible to apply this model to short documents or to sentence compression where redundancy of information is scarce.

Our approach is the first to propose an unsupervised Seq2Seq model where the summary generation is conditioned on the information previously seen by the user. This method is not entirely abstractive; rather, it generates a compressed version of the same input sentence. Nevertheless, it allows us to demonstrate the applicability and portability of our model on new tasks such as the update of short text summarization where training data is scarce.

3. Proposed model

This section presents the general architecture of our suggested approach, as depicted in Fig. 2. As seen in the figure, our model is composed of two competing autoencoders. The first autoencoder learns a language model (LM) to produce coherent texts, and the second autoencoder constrains information (CM) to force the model into create a shorter selection of relevant words to include in the summary.

In Section 3.1, we begin by introducing the classic architecture of the autoencoder used for both models. Then, in Section 3.2, we present the design of our CM, and place special emphasis on the formula that characterize word importance based on their TFIDF scores. It also details how that new input weights the score based on the frequency of previously generated terms. In Section 3.3, we explain the modification made to the LM encoder so that it can consider information from previous iterations when reconstructing the current text. In Section 3.4, we go in greater depth on our personalized objective functions that mimic the behavior of maximum margin relevance approach as well as on how we account for the information constraint. Finally, in Section 3.5, we detail how the two models compete against each other in order to produce a short, relevant and coherent summary at each iteration and how our new parameters influence novelty or consistency in the final outputs.

3.1. Model background

In this paper we use the classic encoder–decoder architecture for our documents. We introduce in this section the basic Seq2Seq model on which we rely. Let us note $T = \{T_1, \dots, T_i, \dots, T_c\}$ the corpus of c documents covering an event across time. Each document T_i corresponds to a specific iteration or update of the news story, and can be represented by a set of N_i words $X_i = \{x_1, x_2, \dots, x_j, \dots, x_{N_i}\}$. The model encoder produces, through the application of a bidirectional Gated Recurrent Unit (GRU) [30], a document encoding D_i and some encoder hidden states $h_1, h_2, \dots, h_j, \dots, h_{N_i}$. When decoding, we perform N_i decoding step to generate our sentence. We start by fixing the initial hidden state of the decoder s_0 to the hidden representation of the document D_i, and, at each decoding step t, a simple GRU decoder estimates the current hidden state s_t with the states s_{t-1} and the predicted word x'_{t-1} at preceding steps:

$$s_t = GRU(s_{t-1}, x'_{t-1}) \tag{1}$$

At decoding step t, the energy vector e_t^j of each input words j and the attention distribution a_t are calculated as in [31]:

$$e_t^j = v^\top \tanh(W_h h_j, W_s s_t, b_{attn}) \tag{2}$$

where v, W_h, W_s, b_{attn} are learnable parameters of the model. We thus obtain an energy vector e_t over the whole input text for each decoding step. The attention is then estimated as a normalized distribution of this energy:

$$a_t = softmax(e_t) \tag{3}$$

Once the attention is calculated, each individual attention value a_t^j is used to weight the representation h_j of each word, allowing the model to create a contextual representation c_t that focus on specific words at each step.

$$c_t = \sum_j a_t^j h_j \tag{4}$$

The context vector is concatenated with the decoder state and passes through a linear and a softmax layer to compute the probability of generating the output word $p_g(x'_t)$:

$$P_g(x'_t) = softmax(W'(W[s_t, c_t] + b) + b') \tag{5}$$

where W', W, b, and b' are learnable parameters. We finally use a copy mechanism as presented in the *Pointer Generator Model* (PGN) [32] to consider Out-Of-Vocabulary words. The new probability of generating the output word x'_t becomes:

$$p_{gen} = \sigma(W_{hgen}^\top c_t + W_{Sgen} s_t + W_x x'_{t-1} + b_{pgen}) \tag{6}$$

$$P(x'_t) = p_{gen} \times P_g(x'_t) + (1 - p_{gen}) \times \sum_{j : x'_j = x'_t} (a_t^j) \tag{7}$$

where σ is the sigmoïd function, W_{hgen}, W_{Sgen}, W_x, and b_{pgen} are learnable parameters. The model is trained with a standard negative log likelihood loss to optimize the generation probability distribution $P(X'_i)$ of the predicted document T'_i, based on the reconstruction of the original input set X_i.

3.2. Informative constraint model

In the context of unsupervised single document summarization, the output should respect a length constraint represented by compression ratio $\alpha < 1$. If we directly apply this constraint to the Seq2Seq autoencoder, the model would simply produce the input αN_i first words of the input. Therefore, we need to implement an additional constraint to identify relevant information to preserve in the summary. Several methods using denoising, topic information, or word probability [12,13,18] are effective in constraining the model to produce a shorter output that respects this specific information. The well-known TFIDF metric emphasizes the relative contribution of terms within a text by counting their occurrences and their dispersion throughout the documents. This metric brings forward the specific information of a document and has proven to be efficient in many natural language processing tasks, especially for news stories that include events with salient features. In our approach, we aim at summarizing the current iteration text T_i which can be expressed as a set of words $X_i = \{x_1, x_2, \dots, x_j, \dots, x_{N_i}\}$ of size N_i. Based on the training dataset, we pretrained a *TFIDF* model, thus attributing a score for each word composing T_i. To create an information constraint, we create a second input for the model where we remove $1 - \alpha$ of the terms from T_i, those with the lowest *TFIDF* score, while preserving its token order. We call this new input the information constraint vector $TFIDF_i = \{x_1, \dots, x_j, \dots, x_{\alpha N_i}\}$, where j still corresponds to the index associated with the original position of the term x in T_i. For example, if we define a sentence $S =$ "The city emergency service confirmed a train accident", vectorized as $T_S = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$. Depending on the pretrained TFIDF model, assuming a ratio $\alpha = 0.4$, a possible constraint sentence could be "emergency confirmed accident". Therefore, we would have $TFIDF_S = \{x_2, x_4, x_7\}$. When testing our model, we replace each Out-Of-Vocabulary word by an unknown token $< UNK >$ indexed in the vocabulary. We specify the value of these tokens to the mean *TFIDF* of all the sentence terms to be able to include these terms in the summary. The model then must learn to encode and decode the two texts simultaneously, thus learning a language model and a vector implicitly defining the size of the summary and the material to preserve. Moreover, to account for text consistency and update information, we add two major modifications to the CM. Following the principle introduced by Luhn [33], we apply a contextual window of size 2 and iteratively go through our text T_i. For each word belonging to $TFIDF_i$, neighbors' scores are multiplied by 1.25. We chose this value empirically after multiple tests because it promotes grammatical contextual words and preserves text coherence while retaining a enough of the original *TFIDF* top score words. These new scores thus increase the local coherence of the selected terms to form a new set $TFIDF_i^C$. Finally, to update information through each news iteration, we also consider the frequency of words present in the text generated T'_{i-1} by the model at the previous iteration $i - 1$ to positively or negatively weight the score. The final CM is thus defined as follows:

$$TFIDF_i^u = \max_{w \in T_i} \sum_{j=1}^{\alpha N} \begin{cases} TFIDF_i^C(w_j) & \text{if } i = 0, \\ TFIDF_i^C(w_j) + \lambda\beta \times TF_{T'_{i-1}}(w_j) & \text{if } i \geq 1 \end{cases} \tag{8}$$

where $TFIDF_i^u$ is the final set composed by the best scoring words for the summary; β is a parameter compensating the frequency of update terms and set as the mean *IDF* score of all terms, here 0.7; and λ is our consistency/novelty parameter that we will discuss later in Section 3.5.

3.3. Dual encoder

The second modification we made to the base model to account for update information is to input the representation of the previous produced text T'_{i-1} when predicting T'_i. More specifically, if the current text to summarize T_i is the set of words $X_i = \{x_1, \dots, x_j, \dots, x_{N_i}\}$, the previous generated text T'_{i-1} can also be defined as the set $X'_{i-1} = \{x'_1, \dots, x'_j, \dots, x'_{M_{i-1}}\}$, where M_{i-1} is the size of the collection of terms generated at the previous iteration. At first iteration $i = 0$, we input an empty set for the summary. Therefore, the model acts as the regular autoencoder introduced Section 3.1. Then, when $i \geq 1$ we provide both text T_i and T'_{i-1} to the encoder to obtain the document representation D_i, previous representation D'_{i-1}, and the two encoding hidden states $h_1, \dots, h_j, \dots, h_{N_i}$ and $h'_1, \dots, h'_j, \dots, h'_{M_{i-1}}$. We concatenate that information to obtain the final vector and set:

$$D_{T_i} = [D_i; D'_{i-1}] \tag{9}$$

$$H_i = \{h_1, \dots, h_{N_i}, h'_1, \dots, h'_{M_{i-1}}\} \tag{10}$$

The new concatenated representation D_{T_i} and all encoding hidden states are provided to the decoder to estimate states and context vectors at each step t. It thus allows the decoder to focus on words from both the current text and previous text when generating T'_i.

3.4. Loss function

Once the information constraint and the update are provided as additional input to the model, we need to set up an objective function that takes them into account. The purpose here is therefore twofold since the model must learn to respect the information in the CM and to follow a LM, allowing it to reconstruct the original texts properly. The goal for the CM is to predict a sequence $TFIDF_i'^u = x'_1, x'_2, \dots, x'_{\alpha N}$ such that we minimize the loss reconstruction related to the CM, which is defined by the cross entropy between this predicted vector and the original $TFIDF_i^u$:

$$\mathcal{L}_{TFIDF}(\theta) = \sum_{x \in TFIDF_i^u} \log p(x'|x;\theta) \tag{11}$$

where θ are the model parameters. To ensure that the result generated by the model $TFIDF'^u$ following the information constraint remains consistent with the input texts T_i, we add a similarity loss between their representation. As in the case of [11], we re-encode the result $TFIDF'^u$ to obtain document representation $D'_{TFIDF'^u}$, and we measure its cosine similarity with the initial D_{T_i}:

$$\mathcal{L}_{COS} = -(1 + cosine\_sim(D'_{TFIDF'^u}, D_{T_i}))/2 \tag{12}$$

Since the cosine similarity can vary from -1 for opposite vectors to 1 for similar ones, we modify the loss in order to obtain normalize values from 0 to 1. This modification pushes the model to obtain a maximum similarity between the two representations. As for the LM, it is possible to increase the lexical and semantic diversity produced by a Seq2Seq model by adding a redundancy condition in its objective function [10]. In the case of update summarization, we thus train the model to account for the content of the preceding text it has generated. Therefore, the objective function becomes:

$$\mathcal{L}_{\mathcal{LM}}(\theta) = \sum_{x \in T_i} \log p(x'|x;\theta) + \lambda \sum_{x_{prev} \in T'_{i-1}} \log p(x'|x_{prev};\theta) \tag{13}$$

where $\lambda \in [-1, 1]$ is a control parameter that makes it possible to consider the update information. Once again, we will further demonstrate and discuss the influence of this control parameter in the generation of summaries by the model in the evaluation Section 5. The final objective of the model is therefore to minimize the total loss:

$$\mathcal{L}_{TOT} = \mathcal{L}_{TFIDF} + \mathcal{L}_{\mathcal{LM}} + \mathcal{L}_{COS} \tag{14}$$

3.5. Generation of novel or consistent summaries

During training, the lambda parameter lets us control information in the CM and enrich the LM in such a way that we can alternate between novelty and consistency between iterations.

- When $\lambda < 0$, Eq. (8) shows that the TFIDF scores will be penalized by the words occurring in the previously generated text. Then, regarding the loss function as defined in (13), we have the equivalent of the MMR approach. In this configuration the model attempts to minimize the likelihood of the previous summary. This is the analog of penalizing the similarity with the term distribution of the previous material. However, this adversarial approach can lead to some instability when $\lambda \leq -0.5$.
- When $\lambda > 0$, Eq. (8) once again shows that the TFIDF scores will be increased when words from previously generated text are repeated in the new summaries. Moreover, regarding the loss function in (13), the model tries to optimize the likelihood of both current and previous texts, thus preferring to include already seen content. In this configuration, it will enforce consistency between each produced update.

At each step, we provide the texts T_i, T'_i, and the updated constraint representations $TFIDF_i^u$ as input to the decoder to respectively output a probability $P(x'_{T_i})$ for the LM and $P(x'_{TFIDF_i^u})$ for the CM. The summary is produced by maximizing the probability of the sequence of words such that:

$$P(x \in S_i;\theta) = \max_{x \in V}[P_{T_i}(x';\theta); P_{TFIDF^u}(x';\theta)] \quad \text{w.r.t } len(S_i) \leq L_S \tag{15}$$

where V is the entire training vocabulary, and L_S is the maximum expected size of the summary S_i for the input text T_i. L_S is set such that $L_S = \alpha N_i$, where α is the compression ratio of the model as defined previously for the CM. At each step, the RNN decoder uses the previous generated word as an input for both models. This shared input makes both models start from the same input, while this competing approach allows us to select the most appropriate term with respect to either the information relevance of the CM or the coherence of the LM.

4. Experimental setup

4.1. Dataset

To evaluate our approach of update summarization of short simple documents, we use the 2013, 2014 and 2015 sections of the TREC track on temporal summarization from the KBA Stream Corpus [34]. The dataset is composed of a set of documents answering a query on a specific event, created using hourly crawls. The task normally consists of extracting representative elements to constitute a summary of each update. These summaries are then compared to reference nuggets manually extracted from the same pool by expert assessors. We note in the dataset that many pairs (e.g. of news, nuggets) come from the same source, and are identifiable through the IDs of the documents provided in the dataset. Therefore, the dataset can be modified to effectively track the information issued by a single origin over time. Once we merge all three datasets, we obtain 6,186 news story for 10,839 text pieces. For our task, a news story then consists of a time series of text, composed of potentially multiple iterations, emitted by a single source of information. We perform several processing operations on the dataset, such as cleaning up URLs, the document ids, or the encoding problems that appear. Moreover, we filtered the news stories with fewer than 5 words and more than 100 words to reduce the noise in the input data. For computational reasons, we also omitted stories longer than five iterations. After applying these filters, we obtained 5,614 document streams, 69% of which contain only 1 text and are thus not subject to any updates, 17% of which have 2 texts, and 12% of which have 3 or more texts. The average size of a document issued from a news story is 38 tokens, and 16 for their associated summaries. We randomly split the dataset with a proportion of 70%, 20%, and 10%. In so doing, we obtained a total of 6,126 examples for training, 1,450 for validation, and 864 for final model testing.

4.2. Evaluation metrics

We first use the F1-ROUGE metrics for ROUGE-1, ROUGE-2 and ROUGE-L [14], which are standard for document summary evaluation. They respectively assess word overlap, bigram overlap, and the longest common subsequence between our references and the summaries generated by our model. We also complement our study with

an unsupervised automatic metric: SUPERT [15], which measures the similarity of content embedded in the summary with the input text. This choice was made to address two major defects that stem from the fact that references are originally created for sets of documents. First, the references can be noisy, providing information from other news sources, and second they can be longer than the input. In light of these challenges, we evaluated our model using this metric, which fits particularly well with unsupervised approaches of text summarization. However, these evaluation metrics only shed light on our model's content relevance. To remedy this issue, we completed the assessment of our model with a human evaluation procedure to consider the grammatical quality of our model. Finally, we conducted several other classic analyses, such as an ablation study, or a sensitivity analysis, and we designed experimental protocols for understanding the novelty and consistency behavior of our approach. The detail of each experiment is detailed in Section 5. The purpose of these analyses is not to evaluate the quality of the model, but to help us achieve a clearer portrait of our model as a whole.

4.3. Implementation

For our experiments, our model uses the GloVe 100 dimensional pre-trained word embeddings (version: glove.6B.100d) [35]. Both the encoder and the decoder of the model are composed of a single bidirectional layer with a size of 512 hidden units. We opted against using a more advanced architecture based on transformers and rather adopted a simpler architecture to better respond to the objectives of this study, which is to focus on the λ parameter controlling the information. The simpler architecture makes it possible to reduce the number of parameters, which in turn accelerated and facilitated the learning process knowing that the modification could improve any model structure. Moreover, it reduces the risk that an architecture with too many parameters and too much capacity may implicitly capture consistency- or novelty-related information, thus reducing the impact of our parameter and potentially distorting our results and analyses. We initialize the weight of the different layers via a Xavier uniform distribution [36], and we established the dropout of each layer at 0.2. To train the model, we conducted a Bayesian optimization of all hyperparameters based on the validation collection. Specifically, all hyperparameters are optimized by a defined-by-run strategy with the Optuna framework [37]. More specifically, we define a hyperparameter space and Optuna seek the minimization of our objective function. We have then selected the best set of hyperparameters after 20 epochs to run our full implementation. Consequently, we train the model with the Adam optimizer [38] with a learning rate of 10^{-4}, a weight decay of 8^{-3} and a gradient clipping of 10. To allow the algorithm to learn to produce good quality texts, we first train the model without accounting for previous iterations. We start updating the CM and LM losses with updates at epoch 50 and we train the model for 80 epochs. Finally, we train the model with stochastic gradient descent using mini batches of dynamic size corresponding to the number of iterations of each news story. We then use the gradient accumulation technique to obtain a final equivalent batch size of 128. To generate the summaries, we define a compression ratio $\alpha = 0.4$ corresponding to the average compression rate in our training data. To improve further the output results, we apply the beam search method with a beam size set to five and an n-gram blocking [39] set to avoid trigram repetitions. We implemented our model with the Pytorch library[5] version 1.8.1. and is available on GitHub.[6] The model was trained on a machine with an 8 Gb NVIDIA Tesla P4 graphics card and a 60 Gb 16-core processor.

[5] https://pytorch.org/

[6] https://github.com/florentfettu/update-summarization-production

Table 1
Average results on the TREC 2013–2015 dataset for update summarization of short documents.

Type	Methods	R-1	R-2	R-L	SUPERT
Baseline	First 9 words (F9 W)	22.64	5.26	19.01	10.95
Supervised	SummaruNNer [40]	26.78	20.12	26.34	13.14
	PGN [32]	16.55	9.11	15.94	10.64
Unsupervised	2-g shuf Denoising AE [12]	16.73	**3.16**	14.13	11.98
	SEQ^3 (Full) [13]	15.67	0.835	11.26	13.24
	USUS_sub_03	16.7	2,13	**14.24**	15.64
	USUS_add_03	16.1	2.34	13.99	15.98
	USUS_sub_08	**16.76**	2.74	14.68	15.53
	USUS_add_08	16.05	2.28	13.95	**16.02**

Bold numbers indicate the best results obtained using unsupervised methods. Supervised results are presented for comparison and context.

5. Results and discussion

5.1. Model evaluation

To obtain a comprehensive view of our method, we compare our model with several baselines. Our first baseline consists in extracting the first 9 words of each document, reproducing the ROUGE principle. Essentially, this consists in extracting the first sentences, which are often used as reference for news stories [12,40]. Then, we decide to compare our model with other recent abstractive approaches. For unsupervised approaches, since our dataset is composed of short texts (i.e. often one sentence in length), we chose sentence compression models as baselines. Specifically, we use the implementation of the denoising autoencoder with Out-Of-Vocabulary words management as presented in [12] and the SEQ3 [13] consisting of two chained autoencoders that consider both topic information and sentence reconstruction in the compression process. We also report the performance of supervised models SummaRuNNer [40] and Pointer Generator Network (PGN) [32] to analyze their ability to capture updated information implicitly thanks to the provided references. Finally, we report the data from our model under three different configurations for Unsupervised Summarization for Updated Sentences (USUS). USUS_add_03 for $\lambda = 0.3$ and USUS_sub_03 for $\lambda = -0.3$. Table 1 presents the comparative results of our analyses for the ROUGE and SUPERT sores.

We observe that the supervised model Summarunner had the best results. The advantage of this model is that it can capture complex relationships between texts through human references. It is therefore difficult to compare it with our model. However, it is interesting to see that our model is the closest to SummaruNNer in performance and even exceeds the PGN architecture. We hypothesize that the relatively poor performance of the PGN is mainly due to its sensitivity to noise in the standard references, especially when the model learns the copying mechanism, whereas SummaRuNNer rather employs a word dropout technique at input level, making it more resilient to this phenomenon. The F9 W baseline, which is created by extracting the 9 first words of each input sentence, is the second-best performing model on the F1 ROUGE score. However, we argue that this score is artificially increased by the summaries' size rather by content salience alone. First, we note that 28% of our input texts have fewer than 9 words after removing stop words. Therefore, the whole input acts as generated summary when computing the score with the reference. The score then reflects the correlation between input and gold nuggets. This hypothesis is corroborated by the SUPERT metrics where the F9 W is outperformed by most models showing that it embeds less salient information from the document. Regarding our model, all our configurations (including our baseline that does not consider information update ($\lambda = 0$)) fairly compared with other unsupervised sentence compression algorithms for the ROUGE metrics. However, the contribution of our approach is clearly evidenced by the SUPERT metric analysis, where our model

Table 2
Human evaluation. Mean scores for 5 native English evaluators.

Methods	Coherence	Content	Redundancy	Consistency
Human references	4.4	3.7	2.16	3.92
2-g shuf [12]	**2.83**	2.74	2.34	2.72
SEQ^3 (Full) [13]	2.71	2.74	2.67	2.74
USUS_add_03	2.63	**3.22**	2.37	**3.27**
USUS_sub_03	2.69	3.13	**1.98**	2.96

outperforms the other unsupervised methods. This once again demonstrates the usefulness of TFIDF for embedding salient information from a news event.

It is important to emphasize here that some limitations exists when evaluating produced summaries with automatic metrics such as ROUGE and SUPERT, especially when the objective is to gauge their quality. Following the work in [12], we conducted a human evaluation of our generated summaries to judge their fluency and the nature of the embedded content. We asked 5 native English speakers to assess the results generated from four models: USUS_sub_03, USUS_add_03, SEQ^3 (Full) [13], and 2-g shuf [12]. We complement the results with the human references to have a fair baseline to compare the models. Each reviewer received a file containing the 5 shuffled summaries associated to 50 randomly selected texts from our test dataset. The evaluators were then instructed to consider two criteria when assessing the produced texts. The first criterion refers to the coherency of the text, which consisted in the summary only. The second criterion refers to the information quality embedded from the source. Here, the file included the original text accompanying the summaries. The evaluators were instructed to rate the summaries on a scale of 1 to 5 for each criterion. Table 2 presents the average evaluation of the summaries' coherence and content. A text receiving a score of 1 in both columns indicates that the summary has poor grammar and encompasses little of the original content.

The results corroborate our initial analyses, in particular the SUPERT score. According to our evaluators, our two models obtain the most relevant results when compared to the baselines. However, as we expected, the semi-extractive capability of our model creates a significant loss of grammaticality for our generated sentences when compared with human references. However, our method's results remain equivalent to other unsupervised abstractive methods. These results can be explained largely due to the combination of the LM, which acted as intended, and the modification of our TFIDF constraint to increase the score of neighboring terms to favor grammatical summaries.

As a result of this study, we observed another emerging trend. Our model USUS_sub_03 performs better on ROUGE while, regarding human evaluations and SUPERT score, USUS_add_03 encompasses more information from the source. As such, our two versions must produce text with different content. Our hypothesis is that USUS_sub_03 favors term novelty between two iterations, while USUS_add_03 enforces consistency between texts. However, it is not possible to draw firm conclusions from these results at this time given the study's limitations; that is, it is not possible to adequately evaluate text quality using only the scores obtained through ROUGE, SUPERT and human evaluators. Indeed, the dataset constitution, where gold references were issued from multiple documents, created some noise. The wrong associations between source texts and summaries may artificially increase the results of our novelty model since the summary may have no relation to its input, and therefore no relation to the previous iteration. In the case of SUPERT, the fact that the texts come from individual sources probably strengthens the impression of consistency between two updates. These factors might explain the performance differences of our two algorithms. To address these issues, we also analyzed the novelty and consistency of the generated results with automatic metrics and human evaluation.

Table 3
Analysis of the composition of update summaries.

Methods	% re-use of new terms from T_i	% re-use of terms from S_{i-1}
2-g shuf Denoising AE [12]	0.33	0.958
SEQ^3 (Full) [13]	0.144	0.936
USUS_sub_03	0.819	0.562
USUS_add_03	0.801	0.837
USUS_sub_05	0.874	0.478
USUS_add_05	0.747	1.161
USUS_sub_08	0.914	0.152
USUS_add_08	0.73	1.242

5.1.1. Novelty and consistency evaluation

To demonstrate the influence of the λ value on the embedded information in the summaries and the novelty/consistency capacities of our model, we set up two automatic metrics to characterize the iterative outputs and the terms employed. The first metric reports the proportion of words that are reused from the source T_i in the summary S_i and that are not present in the previous iteration S_{i-1}. Our second metric measures the ratio of the number of terms in common between S_i and S_{i-1} compared to the same number for T_i and T_{i-1}. When we push for novelty, we expect the information reused from the source to increase while the information from the previous iteration to decrease, and vice versa when we push to produce consistent texts. We then proceeded to a sensitivity analysis of our model with several lambda values and the results for update summaries, where $i > 0$, are reported in the Table 3.

These results highlight two interesting phenomena. The lower values for the direct re-use of source terms obtained in the comparative models [12,13] illustrate their greater capacity of abstraction, but also shows a higher incidence of hallucinations of words not found in the original documents. As for re-using terms coming from the previous summary, the results, close to 1, exhibit that the distributions of terms between two iterative summaries and two iterative input texts are respected and are thus not considered by those algorithms. Our approach also emphases different characteristics for the re-using terms. As the λ value increases, the re-use of terms from the previous iteration increases while the terms from the current source decreases. The opposite phenomenon is observed when we decrease the lambda value. It demonstrates the impact of the λ parameter on the generated output. If there is an increase in the rate of terms re-used from the previous iterations, the model should increase the consistency of the text; if it decreases while favoring the use of terms in the current iteration, then the model should reinforce the novelty. This observation tends to follow the hypothesis that the novelty implementation avoids information overlap, which in turn reinforces the relevance of the results to a user [1], while consistency reinforces the understanding of the source information content [25].

We completed this study by instructing the human evaluators to consider two additional criteria when assessing the produced summaries. Using the same random sample of 50 abstracts, we gave them a file indicating the chronology of the summaries' updates for each event. The first criterion related to the update's redundancy with respect to the summary of the previous iteration. The second criterion concerned how closely the update summary was related to the given event described in the previous iteration. The evaluation was once again conducted using a scale of 1 to 5. Table 2 illustrates the results for different models and the human references. We observe that our USUS_sub_03 model with a negative value of λ produces texts with a lower redundancy than at baseline As for the USUS_add_03 model with a positive value of λ, we note that the human evaluators could more clearly gauge that the two iterations reported the same event. These results confirm that our two approaches can modify the content of the generated summaries.

We can further observe this difference with the example provided in Table 4, produced by our models USUS_sub_03 and USUS_add_03.

Table 4
News stories on Russian elections. The event is updated 3 times. The example shows the summaries generated by the two models USUS_sub_03 and USUS_add_03. The table contains only filtered texts without stopwords and other grammatical terms to focus on information content. New or modified information that appears in subsequent iterations in the two models as a result of the parameter has been *italicized*.

	Original text	Summary $\lambda = -0.3$	Summary $\lambda = +0.3$
0	shot russian elections last sunday video one many examples alleged election fraud went viral started antigovernment protests russia	many russian *election* fraud went viral started antigovernment	many russian *election* fraud went viral started many
1	russian mass protests election results scheduled saturday 30000 people allowed gather moscow s bolotnaya square 11 cities russia also received official permits	russian *mass protests* scheduled saturday 30000 allowed gather cities	russian *election* results scheduled *saturday* 30000 allowed gather russia
2	anti putin activists promoting countrywide protests next saturday suggest shifting internationally hosted websites facebook opposed local social network v kontakte	promoting *putin activists* shifting internationally hosted websites kontakte	promoting *saturday suggest* shifting internationally hosted websites kontakte

These examples demonstrate the influence of the lambda parameter on the final production of summaries. Indeed, in the first iteration, we notice that the word "election" is re-used in the model that pushes for consistency while "mass protest" is favored by the novelty version. The observation is even more telling for the second instance, where the term "saturday" is employed by the consistent variant whereas it is non-existent in the original text of this iteration. Similarly, whereas "putin", a word that does not appear in the first iterations, is pushed very early in the sentence by the novelty model. This analysis of our produced summaries and both automatic and human metrics allow us to safely conclude that our proposed method is indeed able to promote the novelty or the consistency of information, depending on the choice of parameters.

5.2. Ablation study

We conducted an ablation study on our consistent model with $\lambda = +0.3$ to further demonstrate the contribution of the different elements in our model. We place a particular focus on analyzing the implication of adding the summarization information in the dual encoder, the lambda parameters in both the language model and the CM. We first removed the summary in the encoder, thus depriving the system of that information. We observed that the model was unable to continue learning to reconstruct the text. Once it has to consider iterations in the loss function, the decoder experiences a gradient explosion because it does not have enough information to follow the update objective. This demonstrates the critical role of our dual encoder in the model but also its potential instability.

- When $\lambda = 0$ for both models, the embedded information stays relatively similar, with a SUPERT score of 15.9 and a ROUGE1 F1 score remaining at 16.01. However, However, the re-use rates of terms from T_i is 0.8 and S_{i-1} is 0.624, indicating that the model is no longer able to manage the re-use of terms from previous iterations. This shows that now we only aim to maximize the likelihood of the current text and the CM is equivalent to a simple TFIDF optimization. In this way, the approach is similar to existing methods for sentence compression.
- When $\lambda = 0$ for the CM only, we note the same behavior and performance as for the previous configuration where all values set to 0. This provides further evidences that the model relies heavily on the TFIDF metric to produce updated information.
- When $\lambda = 0$ for the LM only, it results in a difference between the two rates of term re-use in the summaries, namely, 0.723 for new terms from T_i and 1.234 for terms from S_{i-1}, which is equivalent to the highest rate observed for $\lambda = 0.8$. However, there is a significant drop in performance for ROUGE and SUPERT—almost 2 points each. This trend seems to confirm that without the LM, the model overfits the TFIDF CM and no longer produces as relevant and coherent outputs.

5.3. Conclusion on the results analysis

The automatic and human analyses demonstrate that the model obtains satisfactory results, which are either equivalent or superior to existing state-of-the-art models in terms of preserving information from the originals texts. Additionally, despite its semi-extractive capabilities, our model produces results that remain understandable and sufficiently coherent for those fluent in English. Furthermore, as evidenced by the experiments, this study shows that it is possible to instruct the model to generate summaries that prioritize either content novelty or content consistency. The ablation study confirms the importance of the lambda parameter in both language and constraint models to deal with iterative information. These results fully highlight the interest of methods that consider updating information where documents are temporally correlated, such as TREC news streams.

6. Conclusion and future work

In this paper, we presented an unsupervised autoencoder method for semi-extractive document summarization. It relies on two competing models that either generate coherent text or control the information included in a summary. By defining explicit constraints and objective functions, we were able to introduce parameters that account for novelty and consistency of information through iterative and/or streams of texts. Therefore, the proposed approach addresses the new task of update sentence compression or short update summarization. As a result, the model outperforms state-of-the-art unsupervised abstractive sentence compression systems for specific datasets such as the TREC temporal track. The model also modulates the information present in the final output, making it more flexible and appropriate for meeting specific needs.

Of course, our approach comes with certain limitations that stem from the architecture used to implement the novelty and cohesiveness parameter. First, in terms of the novelty, we introduce an adversarial learning issue in the LM, wcan lead to gradient explosions, which in turn creations instability in the learning process. This restricts our method to small values of lambda for novelty, thus capping the impact on generated summaries. We obtained good results from initial tests with two lambda value, a small value for the LM and a large one for the CM, which opens the door to further study on stabilization techniques that are likely to improve the model's ability to address novelty. Due to our hardware limits but also with our desire to study explicit information control, we are aware of the shortcomings of our approach compared to recent architectures based on transformers. However, we believe that our findings can be applied to these models in order to increase their portability and their ability to be used for various tasks. In future work, we further hope to consider models with more capacity, especially generative variational autoencoders to understand the impact of our parameters on constraining latent representations. We also plan to investigate different possible solutions for our information constraints. Our current approach only considers shallow statistical metrics to emphasize text relevance. However, the addition of linguistic data such as dependency trees, lexical chains, or discourse-oriented

features could both improve the grammaticality of the model and its capacity to identify consistent or novel content between two updates. Finally, it would be interesting to apply our model to the task of update or iterative sentence compression for long text summarization. Since two following sentences in a document have local information correlation, we expect that models accounting for novelty or consistency could allow automatic systems to efficiently reduce the size of such documents. Notwithstanding these limitations, and given the large potential for improvement, this study positions our model as a strong baseline for assessing the performance of upcoming update sentence compression and summarization tasks.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: M. Florian Carichon has received a MITACS Grant for a partnership between the Québec financial institution "Fédération des Caisses Desjardins du Québec" and HEC Montréal. M. Florent Fettu is an employee of the same financial institution "Fédération des Caisses Desjardins du Québec". M. Gilles Caporossi has no competing interests of any kinds.

Data availability

I have shared the link in the article to my data and code on GitHub.

Acknowledgments

We want to thank the anonymous reviewers who contributed greatly to the improvement of this article with their constructive feedback.

We also want to thank the Fédération des Caisses Desjardins du Québec, Adrien Hernandez, François-Xavier Devailly, and Mohammad Esmaeilpour for their support and feedback during this research project.

Funding

This work received financial support from the Fédération des Caisses Desjardins du Québec and MITACS [Grant IT16112]. The funders also provided the machines allowing us to conduct the experiments in a reasonable time but have no other involvement in the conduct of the research.

References

[1] J. Goldstein, V.O. Mittal, J.G. Carbonell, M. Kantrowitz, Multi-document summarization by sentence extraction, in: NAACL-ANLP 2000 Workshop: Automatic Summarization, 2000.

[2] K. Rudra, N. Ganguly, P. Goyal, S. Ghosh, Extracting and summarizing situational information from the twitter social media during disasters, ACM Trans, Web (TWEB) 12 (3) (2018) 1–35.

[3] J. Allan, R. Gupta, V. Khandelwal, Temporal summaries of new topics, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001, pp. 10–18.

[4] P. Bysani, Detecting novelty in the context of progressive summarization, in: Proceedings of the NAACL HLT 2010 Student Research Workshop, 2010, pp. 13–18.

[5] R. McCreadie, C. Macdonald, I. Ounis, Incremental update summarization: Adaptive sentence selection based on prevalence and novelty, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, 2014, pp. 301–310.

[6] C. Kedzie, K. McKeown, F. Diaz, Predicting salient updates for disaster summarization, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 1608–1617.

[7] R. Agrawal, S. Gollapudi, A. Halverson, S. Ieong, Diversifying search results, in: Proceedings of the Second ACM International Conference on Web Search and Data Mining, 2009, pp. 5–14.

[8] J. Carbonell, J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998, pp. 335–336.

[9] J.-Y. Delort, E. Alfonseca, DualSum: a topic-model based approach for update summarization, in: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, 2012, pp. 214–223.

[10] J. Li, M. Galley, C. Brockett, J. Gao, B. Dolan, A diversity-promoting objective function for neural conversation models, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 110–119, http://dx.doi.org/10.18653/v1/N16-1014, URL https://aclanthology.org/N16-1014.

[11] E. Chu, P. Liu, Meansum: a neural model for unsupervised multi-document abstractive summarization, in: International Conference on Machine Learning, PMLR, 2019, pp. 1223–1232.

[12] T. Févry, J. Phang, Unsupervised sentence compression using denoising auto-encoders, in: Proceedings of the 22nd Conference on Computational Natural Language Learning, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 413–422, http://dx.doi.org/10.18653/v1/K18-1040, URL https://aclanthology.org/K18-1040.

[13] C. Baziotis, I. Androutsopoulos, I. Konstas, A. Potamianos, SEQˆ3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 673–681, http://dx.doi.org/10.18653/v1/N19-1071, URL https://aclanthology.org/N19-1071.

[14] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, 2004, pp. 74–81.

[15] Y. Gao, W. Zhao, S. Eger, SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 1347–1354, http://dx.doi.org/10.18653/v1/2020.acl-main.124, URL https://aclanthology.org/2020.acl-main.124.

[16] A. Ghadimi, H. Beigy, Deep submodular network: An application to multi-document summarization, Expert Syst. Appl. 152 (2020) 113392.

[17] A. Ghadimi, H. Beigy, Hybrid multi-document summarization using pre-trained language models, Expert Syst. Appl. 192 (2022) 116292.

[18] C. Malireddy, T. Maniar, M. Shrivastava, SCAR: sentence compression using autoencoders for reconstruction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2020, pp. 88–94.

[19] P. West, A. Holtzman, J. Buys, Y. Choi, BottleSum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3752–3761, http://dx.doi.org/10.18653/v1/D19-1389, URL https://aclanthology.org/D19-1389.

[20] A. Fabbri, I. Li, T. She, S. Li, D. Radev, Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 1074–1084, http://dx.doi.org/10.18653/v1/P19-1102, URL https://aclanthology.org/P19-1102.

[21] B. Zhao, X. Li, X. Lu, Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7405–7414.

[22] W. Zhu, J. Lu, J. Li, J. Zhou, Dsnet: A flexible detect-to-summarize network for video summarization, IEEE Trans. Image Process. 30 (2020) 948–962.

[23] P. Li, Q. Ye, L. Zhang, L. Yuan, X. Xu, L. Shao, Exploring global diverse attention via pairwise temporal relation for video summarization, Pattern Recognit. 111 (2021) 107677.

[24] B. Zhao, H. Li, X. Lu, X. Li, Reconstructive sequence-graph network for video summarization, IEEE Trans. Pattern Anal. Mach. Intell. 44 (5) (2021) 2793–2801.

[25] R. Barzilay, M. Elhadad, Using lexical chains for text summarization, Adv. Autom. Text Summ. (1999) 111–121.

[26] M.B. Mohammed, W. Al-Hameed, Cohesive summary extraction from multi-document based on artificial neural network, in: 2021 7th International Conference on Contemporary Information Technology and Mathematics, ICCITM, IEEE, 2021, pp. 81–87.

[27] G. Kim, Y. Ko, Effective fake news detection using graph and summarization techniques, Pattern Recognit. Lett. 151 (2021) 135–139.

[28] S. Prabhumoye, C. Quirk, M. Galley, Towards content transfer through grounded text generation, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 2622–2632, http://dx.doi.org/10.18653/v1/N19-1269, URL https://aclanthology.org/N19-1269.

[29] C. Lin, Z. Ouyang, X. Wang, H. Li, Z. Huang, Preserve integrity in realtime event summarization, ACM Trans. Knowl. Discov. Data (TKDD) 15 (3) (2021) 1–29.

[30] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder–decoder approaches, in: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 103–111, http://dx.doi.org/10.3115/v1/W14-4012, URL https://aclanthology.org/W14-4012.

[31] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint arXiv:1409.0473.

[32] A. See, P.J. Liu, C.D. Manning, Get to the point: Summarization with pointer-generator networks, 2017, arXiv preprint arXiv:1704.04368.

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

[34] J.R. Frank, M. Kleiman-Weiner, D.A. Roberts, F. Niu, C. Zhang, C. Ré, I. Soboroff, Building an Entity-Centric Stream Filtering Test Collection for TREC 2012, Tech. Rep., Massachusetts Inst of Tech Cambridge, 2012.

[35] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1532–1543.

[36] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[37] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

[38] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[39] R. Paulus, C. Xiong, R. Socher, A deep reinforced model for abstractive summarization, 2017, arXiv preprint arXiv:1705.04304.

[40] R. Nallapati, F. Zhai, B. Zhou, Summarunner: A recurrent neural network based sequence model for extractive summarization of documents, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

Florian Carichon is a Ph.D. student in the department of data science at HEC Montréal and has been a researcher affiliated with the Group for Research in Decision Analysis (GERAD) laboratory since September 2019 . His topics of interest include natural language processing and deep learning techniques, with a special focus on automatic summarization and unsupervised models.

Florent Fettu has a master's degree in data science from HEC Montreal and is currently working in the banking industry at the Fédération des Caisses Desjardins du Québec. He is proficient in developing, testing and deploying highly adaptive machine learning and deep learning web applications to translate business problems into substantial deliverables.

Gilles Caporossi obtained his Ph.D. at Polytechnique Montréal, where he researched computer-aided scientific discovery in graph theory. In 2003, he joined HEC Montreal as a professor, and he has taught algorithmics, complex network analysis, and automated text analysis for more than a decade. He is a member of the Group for Research in Decision Analysis (GERAD). His research focuses on optimization, computer-aided scientific discovery, and the study of the writing process.