Interactively Learning to Summarise Timelines by Reinforcement Learning

Yuxuan Ye and Edwin Simpson
Intelligent System Laboratory
University of Bristol
{yuxuan.ye, edwin.simpson}@bristol.ac.uk

Abstract

Timeline summarisation (TLS) aims to create a time-ordered summary list concisely describing a series of events with corresponding dates. This differs from general summarisation tasks because it requires the method to capture temporal information besides the main idea of the input documents. This paper proposes a TLS system which can interactively learn from the user’s feedback via reinforcement learning and generate timelines satisfying the user’s interests. We define a compound reward function that can update automatically according to the received feedback through interaction with the user. The system utilises the reward function to fine-tune an abstractive summarisation model via reinforcement learning to guarantee topical coherence, factual consistency and linguistic fluency of the generated summaries. The proposed system avoids the need for reference summaries in training and can adapt to preference feedback from individual users. The experiments show that our system outperforms the baseline on the benchmark TLS dataset and can generate accurate and precise timelines that better satisfy real users.

1 Introduction

Notable news events can last for a relatively long time period. To reveal events’ development legibly, timelines are often applied to present sub-events in time order. For instance, before COVID-19, H1N1 swine flu caused a pandemic around the world. Figure 1 shows the timeline of its early stage. Nowadays, influenced by COVID-19, when people try to recall the precautions applied during the last worldwide epidemic, the huge number of relevant news articles makes it hard to sort out even for experts. On the other hand, even if there are available existing timelines such as Figure 1, they often only briefly narrate the development of sub-event series. For example, it is easy to find that some cases travelled to the UK by flights. But it is still hard for UK readers to obtain details about early precautions that are relevant to their direct concerns, which lowers the timeline’s value to those readers.

Figure 1: An example timeline presenting the early stage of swine flu in 2009.

The tricky problem of huge document collections can be solved by summarisation techniques, which scan the corpus and identify informative text in it. However, timelines are a special type of summary that is chronologically structured. Sub-events subordinating to the main event are summarised individually, and have a corresponding time tag. General summarisation methods, therefore, are usually not directly applicable to TLS tasks. Martschat and Markert (2018) introduced a temporal module into prior summarisation methods to generate timeline summaries. Some prior works (Steen and Markert, 2019; Yu et al., 2021) detected sub-events first and produced summaries with assigned dates to constitute a timeline. We take inspiration from them to apply clustering algorithms in our system to detect sub-events.

Most prior TLS methods (Gholipour Ghalandari and Ifrim, 2020; Martschat and Markert, 2018) focused on generating extractive timeline summaries. Their contents consist of raw sentences copied from the input document collection. Although the syntax of each copied sentence is guaranteed to be correct, irrelevant and redundant information can also be carried in the selected sentences. The tran-
sition between sentences can also be harsh due to probable deletion of transition sentences. Abstractive summarisation approaches are appropriate to solve the issue, but prior works have not applied them to timelines, because the benchmark TLS dataset is too small for training an abstractive summariser which is usually based on deep neural network. This motivates us to design a method utilising abstractive summarisation to generate human-like timeline summaries.

In recent years, reinforcement learning has drawn much attention in academia. Given proper reward signals, it can fine-tune the model without reference data, thus preventing the requirement of large training corpora. Furthermore, by designing appropriate interactive scheme and reward function, interactive learning can be integrated with reinforcement learning, which enables the model to be trained towards the goal set by the user.

In this paper, we proposed an abstractive TLS system that can interact with human users and learn from their feedback via reinforcement learning. Our contributions in this paper can be summarised as follows:

1. We introduce interactive learning and reinforcement learning into a new TLS method, which enables the system to generate tailored timeline summaries without the need for a large training corpus. At the core of our proposal is a modular reward function that incorporates keyword and preference feedback.

2. We evaluate neural abstractive summarisation for TLS for the first time, finding that it improves automated and human evaluation metrics but has issues with factual consistency and repeated content.

3. Our experimental results show that interacting with summaries by specifying keywords and pairwise preferences can improve the quality of summaries. Keywords can guide the model to generate specific details, while pairwise preferences contribute to learn the complex requirements from the user.

Our proposed system outperforms the strong baseline TLS method, CLUST (Gholipour Ghalandari and Ifrim, 2020), on the benchmark dataset Timeline17 (Tran et al., 2013; Binh Tran et al., 2013). All experimental code will be released on acceptance.

2 Related Works

2.1 Automatic Summarisation

Automatic summarisation tasks can be categorised into 2 types, extractive summarisation and abstractive summarisation, according to how the output is generated. Research on the former one started earlier. Mihalcea and Tarau (2004) used graph-based ranking algorithms to select a set of important sentences as the summary of input text, which is still a widely used baseline today. In this stage, researchers also explored graph-based algorithms to generate abstractive summaries (Ganesan et al., 2010). With the rapid development of machine learning, computers have made impressive progress on text representation. It enables algorithms to evaluate sentences from various perspectives instead of only similarity, thus deriving better metrics for selecting key sentences. For example, Liu et al. (2015) applied sparse encoding to compute the coverage, sparsity, and diversity of each sentence, and thereby to cover the main ideas of an input document with the minimum number of sentences. Recent works in automatic summarisation often involved Deep Learning. Deep neural networks, such as BERT (Devlin et al., 2019), were widely applied in both summarisation tasks. They can provide strong word embeddings (Zheng and Lapata, 2019) and classify whether the input sentence should appear in the summary (Liu, 2019). Transformer neural networks also empowered language models to learn from huge corpora and generate human-like abstractive summaries (Raffel et al., 2019; Zhang et al., 2020).

2.2 Reinforcement Learning in Natural Language Generation

The application of reinforcement learning on Natural Language Generation tasks has received some success recently. These approaches can guarantee certain necessary properties of the generated output by designing corresponding reward signals. Song et al. (2020) and Mesgar et al. (2020) used reinforcement learning to improve the factual consistency of dialogue systems. Prior works (Gao et al., 2018) on automatic summarisation applied interactive learning to learn reward functions from users. In this case, the human user is not required to directly provide the reward for hundreds of episodes of learning: reinforcement learning agents learn the policy offline using a reward function learned from a small number of user-supplied labels.
3 Background

The proposed method in this paper focuses on adapting an abstractive summariser to TLS tasks. The workflow of detecting sub-events of the timeline follows the baseline, CLUST (Gholipour Ghalandari and Ifrim, 2020), which is one of the state-of-the-art event detection TLS methods. To lay a solid foundation for our system, we recap its necessary details, the improvements we made on top of it, and important concepts of reinforcement learning in Natural Language Generation in this section.

3.1 CLUST workflow

CLUST divides the TLS task into two stages, described as follows:

1. The first stage identifies mentioned dates and detects sub-events from document collections.
   (a) Apply Heideltime (Strötgen and Gertz, 2010), an advanced temporal tagging tool, to annotate all the date expressions in raw documents.
   (b) Compute document vectors by TF-IDF (Jones, 1972), then cluster document vectors by Markov Clustering (Van Dongen, 2000) with temporal constraints.
   (c) Assign a date for each cluster. The assigned date is the most mentioned one in the documents of the cluster.
   (d) Rank all clusters by their importance, which is computed by the number of times the assigned date is mentioned.

2. The system generates summaries for each sub-event and builds the timeline in the second stage.
   (a) Apply the extractive multi-document summarisation method, CentroidOPT (Gholipour Ghalandari, 2017), on the top- $k$ important clusters.
   (b) Combine all generated summaries with their dates to form the timeline.

The document vectors used for clustering are computed by TF-IDF, which is outperformed by dense word embedding on many tasks, such as computing semantic textual similarity. Therefore, we use Sentence-BERT (Reimers and Gurevych, 2019) to embed sentences in the document into 768-d vectors, then compute the average sentence vector as the document vector.

As for the clustering algorithm, we replace Markov Clustering by Affinity Propagation (Frey and Dueck, 2007), as it does not require the user to input the number of clusters, and is widely used in clustering complex and unstructured datasets. To reach better clustering results, we reserve the idea of temporal constraints from CLUST and adapt it to Affinity Propagation. News articles that have close publish dates are more likely to describe the same sub-event. In contrast, articles with distant publish dates, even if they are adjacent in semantic space, should be considered as describing similar but disparate sub-events. For example, a series of aftershocks often happen after the main earthquake. Although many aftershocks may have similar levels, they should be recorded separately by their happening dates in the timeline.

3.2 Reinforcement Learning in Natural Language Generation

Reinforcement learning searches for an optimal solution for Markov Decision Processes. It is commonly used to formulate and model sequential decision making problems, which summarisation can be modelled as. In abstractive summarisation tasks, the summariser is often regarded as the agent. It performs actions by sequentially selecting tokens (words and punctuation) from the vocabulary and appending them to the current generated text. In this paper, we choose PEGASUS (Zhang et al., 2020) as our summariser. Prior works (Peters et al., 2019; Mesgar et al., 2021) have shown that fine-tuning the language head of the model can receive competitive benefits compared to fine-tuning the whole model, and is more computationally efficient. Therefore, we only tune the last linear layer of the model via reinforcement learning, as highlighted in red in Figure 2.

A reinforcement learning problem is defined by a tuple $(S, A, P, R, T)$. Their definitions in this scenario are defined as follows:

- $S$: The set of states. The state $s \in S$ is a length-fixed vector encoded and padded by the tokeniser, which represents the generated summary so far.
- $A$: The set of actions, which is usually the vocabulary. The action $a \in A$ is the token_id in the vocabulary.
- $P$: The transition probability $P(s'|s,a) \rightarrow \mathbb{R}$ representing the probability of be-
Figure 2: Fine-tuning the language head (highlighted in red) of Transformer, adapted from Vaswani et al. (2017).

Figure 3: The structure of the timeline summarisation system.

4 Methods

Our main methodological contribution is to integrate an iterative interaction process into the event summarisation step of the TLS system. To have a clear view of our interactive method’s performance, Figure 3 shows the workflow of the proposed method, which was adapted from CLUST.

4.1 Notations

We denote the generated summary so far as $s$. Tokens in $s$ are marked as $s_i$, i.e., $s = (s_1, s_2, \ldots, s_l)$, where $l$ is the maximum length of the output. The user’s feedback is noted as $M, N, P$, individually representing wanted keywords, unwanted keywords and preferences. $D$ denotes the input documents.

4.2 Interactive Learning

The interaction process starts after the event detection stage is done. It allows the user to preview the draft timeline and provide feedback, as shown in Figure 4.

Currently, the system accepts 3 feedback types, including wanted keywords, unwanted keywords, and preferences between pairs of draft timelines. The user can directly input the two types of keywords and use 0/1 to give the preference label, indicating which of two drafts is preferred. The keyword feedback is designed to receive specific details that the user wants to add or remove in the
current timeline. And the preference label represents more complex requirements from the user, such as rhetoric and writing style, which are difficult to formulate.

The received feedback will be used to update the reward function, which is then used for reinforcement learning to fine-tune the summariser. Then the system can generate a new timeline to start another round of interaction with the user. The system does not require many rounds of interactive learning to achieve the goal of generating tailored timelines for the user because the rewards can be learned from small amounts of feedback then used for many episodes of offline reinforcement learning.

4.3 Reward Functions

The feedback input by the human user is still unreadable for the agent. Therefore, a compound reward function is defined to convert the feedback into numerical signals. Other necessary properties of a good timeline summary are also formulated into signals to guarantee the quality of the summary.

\[ R_1 \] aims to ensure the factual consistency of the summary. In other words, the generated summary should be highly similar to the source document on semantics. To capture the semantic information, we obtain the sentence embedding by paraphrase-DistilRoBERTa-v1 (Reimers and Gurevych, 2019) and use the average sentence vectors to represent the summary and input documents in the semantic space. Prior works have shown that cosine similarity is a reliable metric of factual consistency for abstractive summarisation (Zhang et al., 2019). Therefore, we define \( R_1 \) as follows:

\[
R_1 = \cos \langle \vec{s}, \vec{D} \rangle
\]  

\( R_2 \) contributes to the coherence between the summary and the user's feedback, which automatically updates when receiving new feedback from the user. It is a weighted sum of three functions. The first two functions are the rewards for topical coherence between keywords feedback and the output. They are computed in the same way as \( R_1 \). Here, we use GPPL (Simpson and Gurevych, 2020) as our preference learning model to map pairwise labels to scores reflecting user preference. The weights of the three function sum up to 1 and are chosen by tuning on a validation set.

\[
R_2 = w_1 \sum_{m_i \in M} \cos \langle \vec{m}_i, \vec{s} \rangle - w_2 \sum_{n_i \in N} \cos \langle \vec{n}_i, \vec{s} \rangle + w_3 \text{score}(s, P)
\]  

\( R_3 \) quantifies the linguistic fluency of the summary. \( N(\cdot) \) is the negative log-likelihood loss calculated by a language model. \( \alpha \) is the maximum of \( N(\cdot) \) which normalises the sub-reward function.

\[
R_3 = \frac{\alpha - N(s)}{\alpha}
\]

\( R_4 \) penalises generating repeated unigrams. It is well-known that reinforcement learning agents sometimes generate repeated tokens if they find some tokens receive high rewards. So, we define \( R_4 \) as follows:

\[
R_4 = 1 - \frac{\#\text{repeated_tokens}}{\#\text{tokens}}
\]

\( R_1 \) and \( R_2 \) encourage the system to generate summaries with correct and required content. \( R_3 \) and \( R_4 \) contribute to the linguistic quality of the text. The final reward function is the weighted sum that balances these different qualities:

\[
R = \gamma_1 R_1 + \gamma_2 R_2 + \gamma_3 R_3 + \gamma_4 R_4
\]

4.4 Training Algorithm

The expected reward can be written as:

\[
L = \mathbb{E}_{\pi_\theta} [R(s, (M, N, P))],
\]

where \( \theta \) is the parameters of the policy function \( \pi \). Mnih et al. (2016) gave the gradient of \( \theta \) with respect to \( L \) as follows:

\[
\frac{\partial L}{\partial \theta} = \mathbb{E}_{\pi_\theta} \left[ R(s, (M, N, P)) \frac{\partial \pi_\theta}{\partial \theta} \right]
\]

Mnih et al. (2016) also pointed out that a modified policy gradient method, Actor-Critic, can be applied to fine-tune the policy function \( \pi_\theta \). It not only approximates the policy function but also the state-value function. The advantage function, \( \delta \), is introduced into the algorithm to compute, in current state \( s \), how much better the value of the action \( a \) is than the average:

\[
\delta = R + \gamma \tilde{v}(s', w) - \tilde{v}(s, w),
\]
where $\gamma$ is the discount rate, $\hat{v}$ is the critic function estimating the real value function and $w$ is the critic’s parameter vector. Optimising the parameters $\theta$ by multiplying the advantage $\delta$ with the gradient improves the likelihood of choosing actions with a positive influence on the final reward. Therefore, Actor-Critic can reduce the variance of sampling from the policy distribution.

In this paper, we use PEGASUS as the actor and a three-layer fully connected neural network as the critic. The critic’s parameters $w$ are updated by the squared error between its predicted rewards and the value given by the reward function.

5 Experimental Setup

5.1 Dataset

Timeline17 (Tran et al., 2013; Binh Tran et al., 2013) is the benchmark dataset in Timeline Summarisation field. It contains 19 manually created timelines with their associated news articles. Each timeline and its news documents belong to a famous news agency, such as BBC and FoxNews. The characteristics of Timeline17 are shown in Table 1.

| #Timelines | Ave. #Docs | Ave. #PubDates |
|------------|------------|----------------|
| 19         | 508        | 124            |

Table 1: The Statistic characteristics of Timeline17, cited from Gholipour Ghalbondari and Ifrim (2020).

5.2 Implementation

To have a better comparison with the baseline, we keep the original experiment settings where possible. Our main settings are as follows:

- The length of generated timelines are limited to be shorter or equal to the corresponding reference timeline.

Prior sections have mentioned that PEGASUS is chosen to be the abstractive summariser (reinforcement learning agent). In fact, PEGASUS provides a wide range of versions for users. We choose PEGASUS-Multi_News (Zhang et al., 2020) in the implementation because it is tuned on news datasets from a similar domain to ours.

5.3 Timeline Evaluation Metrics

ROUGE (Lin, 2004) is one of the most commonly used evaluation metrics in summarisation tasks. However, directly applying ROUGE on timeline summaries neglects the influence of temporal information. Martschat and Markert (2017) proposed an adapted ROUGE based on the date alignment mechanism, which is the most commonly used evaluation metric in TLS.

Alignment ROUGE manages to find a mapping from the output timeline’s date to the reference date which minimise the distance of each mapping pairs. Then ROUGE scores are computed on pairwise summaries, and the average of them is used as the final score.

5.4 Simulated Interaction

Since the system requires at least half an hour to finish the training process for a topic, we design a simulated user to provide feedback while developing the system.

**Keywords** We use TF-IDF (Jones, 1972) to represent the importance of each word. 10 Words with the highest TF-IDF values are saved as the wanted keywords list. And 10 words with the lowest values are save as the unwanted list. These keywords are fed to the system in each interaction round.

**Preferences** We pre-generate several versions of timelines for each topic. We rank these timelines by their Alignment ROUGE score. Therefore, we can extract multiple pairwise preference labels from the rankings. The system is able to learn the score function and mark the generated summary as the reward.

6 Results

6.1 Clustering Analysis

In the event detection stage, we replace the clustering algorithm by Affinity Propagation (Frey and
Dueck, 2007). Table 2 shows that the number of clusters generated by Affinity Propagation is closer to the average length of the given timelines. Considering the ratio of the average timeline length and the average number of clusters, Markov Clustering discarded more than 70% clusters, which implies that Affinity Propagation has better granularity on clustering results. Table 3 shows that Affinity Propagation has higher precision, which indicates that the dates selected by Affinity Propagation are more likely to be correct.

|       | Average Clusters | Average Length |
|-------|------------------|----------------|
| MC    | 148.06           | 38.68          |
| AP    | 50.90            | 29.17          |

Table 2: The average number of clusters generated by Markov Clustering and Affinity Propagation. And the average timeline length when applying the two clustering algorithms.

|       | Precision | Recall | F1   |
|-------|-----------|--------|------|
| MC    | 0.48      | 0.42   | 0.45 |
| AP    | 0.52      | 0.29   | 0.38 |

Table 3: The precision, recall and F1 score of date selection of the two clustering algorithms. The ground truth dates are from the reference timelines in the dataset.

### 6.2 ROUGE Score

When evaluating the timeline summaries’ content, we expect that our timelines can correctly cover the main words that appear in the reference timelines. Therefore, ROUGE-F1 score is the most suitable metric, because it reflects the overlaps’ precision and recall in a balanced manner.

Table 4 shows the results of the proposed TLS system and CLUST. $r = 0$ means that the summariser directly summarise the clusters without any fine-tuning. Otherwise, each interaction round is composed of several episodes of reinforcement learning. The zero-interaction setting outperforms the baseline on Alignment ROUGE1-F1 and receives a competitive score on Alignment ROUGE2-F2 indicating that the replacement we made in the event detection stage makes sense. With proper interaction rounds, the system outperforms the baseline and has a huge increase on Alignment ROUGE2-F1 score.

| Interaction Rounds($r$) | Alignment ROUGE1-F | Alignment ROUGE2-F |
|-------------------------|--------------------|--------------------|
| CLUST                   | 0.0763             | 0.0185             |
| $r=0$                   | 0.0824             | 0.0195             |
| $r=1$                   | 0.0813             | 0.0182             |
| $r=2$                   | **0.0838**         | 0.0197             |
| $r=3$                   | 0.0794             | **0.0219**         |

Table 4: Results of simulated reinforcement learning based interactive TLS system. The agent samples multiple episodes and learns from them over all clusters in each interaction round.

### 6.3 User Study

To further understand the performance of our system, we looked into the content it generated and compare it with the baseline’s output. The comparison is attached in Appendix A due to the page limit. Besides the numerical evaluation and the content analysis, we also invited real human users to rank the timelines. All the participants are asked to read the timelines generated by the system and rank them from 4 perspectives:

- **Relevance** How relevant the content is to the topic.
- **Redundancy** How much information is repeated in the timeline.
- **Text Legibility** How easy it is to read the timeline.
- **Informativeness** How much information does the timeline contain.

Since evaluating text quality is complex and subjective in some cases, an overall ranking is also required, where the participant can use their own standard to evaluate the timelines. Table 5 shows the experimental results from 10 participants.

The user study shows that the timeline generated by $r = 2$ system receives the highest ranking, indicating that our algorithm improves the system’s performance with proper rounds of interaction. $r = 1$ and $r = 3$ have advantages in different aspects. The relevance reveals that one interaction round is not sufficient to instruct the summariser to generate content that satisfies the user’s interests.

However, after learning from too many episodes, the system still starts to trick the reward function, even though we implemented repeated token penalty via Equation 4. It generates repeated
Interaction Rounds \((r)\) 
\[ \begin{array}{lcccc}
    r = 0 & r = 1 & r = 2 & r = 3 \\
    \hline
    \text{Informativeness}\,* & 2.57 & 2.21 & 2.29 & 2.93 \\
    \text{Text Legibility} & 2.75 & 2.32 & 2.32 & 2.61 \\
    \text{Redundancy} & 2.18 & 2.54 & 2.46 & 2.82 \\
    \text{Relevance}\,** & 2.93 & 2.79 & 2.07 & 2.21 \\
    \text{Overall}\,** & 2.82 & 2.50 & 1.96 & 2.71 \\
\end{array} \]

Table 5: Average rankings for the timelines given by the real human users. Overall ranking and Relevance (**) have significant difference \((p < 0.05)\) among different interaction rounds. Informativeness (*) is at the margin of statistical significance \((p = 0.069)\). Redundancy is near-marginal significance. And Text legibility seems to have no significant difference.

Table 5: Average rankings for the timelines given by the real human users. Overall ranking and Relevance (**) have significant difference \((p < 0.05)\) among different interaction rounds. Informativeness (*) is at the margin of statistical significance \((p = 0.069)\). Redundancy is near-marginal significance. And Text legibility seems to have no significant difference.

Table 5: Average rankings for the timelines given by the real human users. Overall ranking and Relevance (**) have significant difference \((p < 0.05)\) among different interaction rounds. Informativeness (*) is at the margin of statistical significance \((p = 0.069)\). Redundancy is near-marginal significance. And Text legibility seems to have no significant difference.

Table 5: Average rankings for the timelines given by the real human users. Overall ranking and Relevance (**) have significant difference \((p < 0.05)\) among different interaction rounds. Informativeness (*) is at the margin of statistical significance \((p = 0.069)\). Redundancy is near-marginal significance. And Text legibility seems to have no significant difference.

Table 5: Average rankings for the timelines given by the real human users. Overall ranking and Relevance (**) have significant difference \((p < 0.05)\) among different interaction rounds. Informativeness (*) is at the margin of statistical significance \((p = 0.069)\). Redundancy is near-marginal significance. And Text legibility seems to have no significant difference.

Health officials around the world are keeping a close eye on the swine flu outbreak that has killed at least 60 people in Mexico and spread to the US. Here are some questions and answers you should ask yourself: How many people are affected? Mexico has confirmed 20 deaths from the flu and has 40 other possible fatalities and 1,004 people reported infected, according to the AP. How many people are in the US? Seven people have been infected in the state of California, including two high-school students who recently returned from Mexico, the Los Angeles Times reports. How many people could be infected in the US? There are currently no confirmed cases in the US. How many people could be infected in Mexico? There are currently no confirmed cases in Mexico.

Figure 5: The example of highly rewarded bad summary. Red sentences and green sentences are conflict facts. Blue sentences are slightly varied from each other.

Besides the timeline rankings, we required the participants to identify whether they are native English speakers in experiments. Another interesting finding is that native speakers focus more on the informativeness, while nonnative speakers attach importance to conciseness so that they can easily find the important information nuggets. This may cause them to rate the timelines differently, indicating that it is a noteworthy aspect to design the reward function in future work.

7 Conclusion

We propose a new reinforcement learning based interactive learning timeline summarisation system, which is able to generate tailored timeline summaries according to the user’s interests without referring to the golden standard data. Our system learns from both preference labels and keyword feedback, which enables the system to learn both topical and holistic preferences. We design a compound reward function for the reinforcement learning phase to maximise the output quality. It is designed to automatically translate the user’s feedback into summary evaluation functions, enabling the system to leverage small amounts of user feedback to guide fine-tuning. We simulated the users’ feedback, and compared the system’s performance against the baseline. Our proposed system outperforms the baseline in the simulation experiments. The further exploration suggests that the reinforcement learning based interaction process changes the summaries’ content towards the user’s interests, and human evaluation results shows that real human users prefer the summaries after fine-tuning.

8 Limitations

Although our proposed system outperforms the strong baseline and receives statistically significant user study results, its limitations are still noteworthy because they indicate potential research directions for future works.

Small user study group Due to that the user study requires the participants to read huge amounts of documents, we only have results from 10 volunteers, which is a very small group. Therefore, although we identified differences between native and non-native speakers, this still needs further investigation.

Languages Currently, we only explored English summarisation. Other languages, such as Chinese and German, have different grammatical structure. Whether our paradigm is applicable to different languages still needs verification.

Simulated user feedback Our simulated experiments demonstrate that the reinforcement learning approach can learn from this kind of feedback, but
future work will test the system with the keywords that users specify in a real deployment.

**Limited scale of the dataset** Due to that generating a timeline dataset is expensive, we can only finish our experiment on a small dataset. Although our findings are statistically significant, it would be more convincing if our work can be verified on a larger dataset.

**Reward function design** In Figure 5, we show that our system still learned to obtain rewards by tricking. It indicates that a more sophisticated reward function is needed to avoid the summariser learning those unacceptable behaviours. And other aspects, such as native speaker, should be added into consideration when designing the reward function as well.

**References**

Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. Predicting relevant news events for timeline summaries. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 91–92.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of deep bidirectional transformers for language understanding*. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Kavita Ganesan, Cheng-Xiang Zhai, and Jiawei Han. 2010. *Opinosis: A graph based approach to abstractive summarization of highly redundant opinions*. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348, Beijing, China. Coling 2010 Organizing Committee.

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2018. *APRIL: Interactively learning to summarise by combining active preference learning and reinforcement learning*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4120–4130, Brussels, Belgium. Association for Computational Linguistics.

Demian Gholipour Ghalandari. 2017. *Revisiting the centroid-based method: A strong baseline for multi-document summarization*. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 85–90, Copenhagen, Denmark. Association for Computational Linguistics.

Demian Gholipour Ghalandari and Georgiana Ifrim. 2020. Examining the state-of-the-art in news timeline summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1322–1334, Online. Association for Computational Linguistics.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Chin-Yew Lin. 2004. *ROUGE: A package for automatic evaluation of summaries*. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

He Liu, Hongliang Yu, and Zhi-Hong Deng. 2015. Multi-document summarization based on two-level sparse representation model. In *Twenty-ninth AAAI conference on artificial intelligence*.

Yang Liu. 2019. *Fine-tune BERT for extractive summarization*. *CoRR*, abs/1903.10318.

Sebastian Martschat and Katja Markert. 2017. Improving ROUGE for timeline summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 285–290, Valencia, Spain. Association for Computational Linguistics.

Sebastian Martschat and Katja Markert. 2018. A temporally sensitive submodularity framework for timeline summarization. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 230–240, Brussels, Belgium. Association for Computational Linguistics.

Mohsen Mesgar, Edwin Simpson, and Iryna Gurevych. 2021. Improving factual consistency between a response and persona facts. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 549–562, Online. Association for Computational Linguistics.

Mohsen Mesgar, Edwin Simpson, Yue Wang, and Iryna Gurevych. 2020. Generating persona-consistent dialogue responses using deep reinforcement learning. *CoRR*, abs/2005.00036.

Rada Mihalcea and Paul Tarau. 2004. *TextRank: Bringing order into text*. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR.
Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pre-trained representations to diverse tasks. In Proceedings of the 4th Workshop on Representation Learning for NLP (RepL@NLP-2019), pages 7–14, Florence, Italy. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Edwin Simpson and Iryna Gurevych. 2020. Scalable bayesian preference learning for crowds. Machine Learning, 109(4):689–718.

Haoyu Song, Wei-Nan Zhang, Jingwen Hu, and Ting Liu. 2020. Generating persona consistent dialogues by exploiting natural language inference. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 8878–8885.

Julius Steen and Katja Markert. 2019. Abstractive timeline summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 21–31, Hong Kong, China. Association for Computational Linguistics.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

Giang Binh Tran, TA Tran, Nam-Khanh Tran, Mohammad Alrifai, and Nattiya Kanhabua. 2013. Leverage learning to rank in an optimization framework for timeline summarization. In TAIA workshop, SIGIR, volume 13.

Stijn Marinus Van Dongen. 2000. Graph clustering by flow simulation. Ph.D. thesis.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems, 30.

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

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

A Output Analysis
To confirm the validity of the proposed method, we look into the generated timeline summaries, as shown in Figure 6(a). Some noteworthy text is highlighted in different colours. As comparison, Figure 6 shows the timeline generated by the baseline in the same time period. It is not difficult to find that the extractive timeline is less legible, because many html control tokens are pasted into it along with the useful information, such as \(-LRB-\) and \(-RRB-\). The timelines generated by baseline and \(r = 0\) system contain the charity activities. But with increasing interaction rounds, the timelines focus more on victims, rescue teams and relief workers that are directly connected to the earthquake. So the proposed system is able to tailor the summaries content according to the user’s interests.
| Interaction Rounds(r) | r = 0 | r = 1 | r = 2 | r = 3 |
|----------------------|-------|-------|-------|-------|
| 2010-01-12           | Wyclef Jean is on his way to Haiti to help victims of the country's devastating earthquake. They need our help ;Please help if you can; the Haitian-borne rapper tweeted. | Bill Clinton has arrived in Port-au-Prince to offer his help in the aftermath of the Haitian earthquake. The former president, who is currently the UN special envoy to the country, met with Haitian President Jean-Bertrand Aristide and other officials today, the AP reports. | As the scope of the Haiti earthquake becomes clear, one thing is becoming clear: The toll is likely to be in the tens of thousands, reports the New York Times. | Relief workers from around the world are en route to Haiti to help in the wake of the country's devastating earthquake. The first foreign rescue team is en route to Port-au-Prince from the Dominican Republic, the BBC reports. |
| 2010-01-19           | The death toll from the Haiti earthquake now stands at 50,000, according to Haitian officials, but that doesn't mean it's a done deal. | As the scope of the Haiti disaster becomes clear, one thing is becoming clear: The nation's infrastructure is in such bad shape that it could take years to rebuild, reports the New York Times. | The death toll from last week's earthquake in Haiti now stands at 100,000, according to the country's president, who says the real toll is closer to 200,000. | The death toll from the Haiti earthquake now stands at 50,000, according to the country's interior minister, and that number is sure to rise. |
| 2010-01-21           | Simon Cowell is planning to put out a charity single to raise funds for Haiti, and he and Cheryl Cole on board. | Relief workers in Haiti are still struggling to get food, water, and medical assistance to survivors as the death toll from the country's devastating earthquake rises to at least 200,000. | As the scope of the Haiti disaster becomes clear, one thing is becoming clear: The toll is going to be a lot higher than previously thought. | Relief workers in Haiti are still struggling to get food, water, and medical supplies to survivors, and the country's interior minister says 100,000 people will initially be sent to 10 settlements near Port-au-Prince. |

(a) The timeline of Haiti Earthquake generated by the system.

(b) The timeline of Haiti Earthquake generated by the baseline in the same period.

Figure 6: Example generated timelines.