ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230601

Towards Abstractive Timeline Summarisation Using Preference-Based Reinforcement Learning

Yuxuan Ye[a] **and Edwin Simpson**[a;*]

[a]Intelligent Systems Laboratory, University of Bristol
ORCiD ID: Yuxuan Ye https://orcid.org/0000-0003-1677-5196,
Edwin Simpson https://orcid.org/0000-0002-6447-1552

Abstract. This paper introduces a novel pipeline for summarising timelines of events reported by multiple news sources. Transformer-based models for abstractive summarisation generate coherent and concise summaries of long documents but can fail to outperform established extractive methods on specialised tasks such as timeline summarisation (TLS). While extractive summaries are more faithful to their sources, they may be less readable and contain redundant or unnecessary information. This paper proposes a preference-based reinforcement learning (PBRL) method for adapting pretrained abstractive summarisers to TLS, which can overcome the drawbacks of extractive timeline summaries. We define a compound reward function that learns from keywords of interest and pairwise preference labels, which we use to fine-tune a pretrained abstractive summariser via offline reinforcement learning. We carry out both automated and human evaluation on three datasets, finding that our method outperforms a comparable extractive TLS method on two of the three benchmark datasets, and participants prefer our method's summaries to those of both the extractive TLS method and the pretrained abstractive model. The method does not require expensive reference summaries and needs only a small number of preferences to align the generated summaries with human preferences. Code available at https://github.com/Haruhi07/PBRL-TLS.

1 Introduction

Keeping up with the news on a topic of interest involves reading multiple articles from various news providers published across a range of dates. Timeline summarisation (TLS) makes this task easier by presenting a chronological list of the main events related to a specific subject, distilled from multiple sources [6, 29]. Table 1 shows examples of timelines generated by extractive and abstractive summarisation. In typical news summarisation tasks, abstractive summarisation can produce summaries of comparable quality to human authors [33]. However, successful pretrained models such as BART [13] and PEGASUS [33] cannot be directly applied to TLS as, unlike the tasks these models were trained on, TLS requires the system to identify key dates and structure the summary into corresponding events.

For specialised summarisation tasks, such as TLS, training data is in short supply, as example summaries are expensive to acquire. Therefore, existing TLS systems [17, 9, 14] often rely on extractive methods [16, 10] that select sentences from the source articles and avoid fine-tuning large neural networks. While extractive summaries are highly faithful to their source articles, concatenating pre-existing sentences may result in summaries that contain unwanted information, repeat earlier points, omit relevant information or lack fluency. While pretrained transformers are a potential solution, they have not been evaluated in previous abstractive TLS systems [27, 3, 4]. We therefore investigate a new approach to adapt pretrained transformers to TLS that avoids the cost of writing reference summaries.

In this paper, we propose to learn a reward function from a combination of pairwise preference labels and keywords provided by human annotators, which is used to directly optimise a pretrained summarisation model by reinforcement learning. This combined approach is known as preference-based reinforcement learning (PBRL) [5]. For a pair of timeline drafts, $s1$ and $s2$, a pairwise label $P(s1, s2)$ indicates that the annotator prefers $s1$ to $s2$. We use a set of preference labels to train a preference model that can evaluate the quality of any given summary [28, 26], and include this preference model into our reward function. Preference models have been shown to be highly consistent with user choices [11], can outperform established metrics of summary quality that depend on reference summaries [35], and can be learned from small numbers of preferences [26]. While the preference model provides holistic guidance to the summariser, keywords directly specify themes that are important to include in the summary. Including keywords in the reward function therefore aims to focus the timeline when summarising a large number of news articles that cover different aspects of the same topic. To produce abstractive timeline summaries, we embed a pretrained abstractive summariser into a TLS pipeline, and fine-tune it using reinforcement learning with our preference-based reward, without the need for any reference timelines. Thus, learning the reward from human feedback aligns the summaries with annotator's preferences [2], which we find results in timelines that are preferred by human evaluators over those of a closely comparable extractive method and a zero-shot abstractive summariser.

The core contributions of this work are as follows: (1) An approach for adapting pretrained summarisation models to TLS without the need for reference timelines, using PBRL with a compound reward function; (2) The first evaluation of a pretrained transformer for abstractive timeline summarisation on three benchmark datasets, finding that zero-shot performance is marginally worse than the closest extractive alternative [9]; (3) We show that timelines produced after fine-tuning with PBRL have higher BERTScores [34] and are preferred to extractive summaries by human evaluators.

* Corresponding Author. Email: edwin.simpson@bristol.ac.uk.

Table 1. Timeline examples of Libya in Arab Spring generated by extractive and abstractive systems.

Ext. Timeline	2011-03-02	What's happening in Libya? Libya holds the largest crude oil reserves in Africa and oil prices rose to a two-year high today.
	2011-03-19	The French and British governments have lead the military intervention in Libya, despite having been among the most enthusiastic supporters of Gaddafi 's rehabilitation in the 2000s [AFP].
	2011-04-20	Tim Hetherington – who was nominated for an Oscar this year with co-director Sebastian Junger for "Restrepo", a documentary about U.S. troops in Afghanistan, and Chris Hondros, a New York-based photographer for the Getty agency – were reportedly killed in the volatile city of Misrata.
Abs. Timeline	2011-03-02	Crude oil rose for a second day in New York on Monday as the crisis in Libya and unrest in the Middle East continued to rattle investors.
	2011-03-19	France's military intervention in Libya has been described as "Sarkozy's war" by the French philosopher Bernard - Henri Levy, who helped to persuade President Nicolas Sarkozy to arm the Libyan rebels.
	2011-04-20	A British film director and war photographer who was nominated for an Oscar has been killed in a mortar attack in Libya.

2 Background

2.1 Event Detection

Prior TLS systems use various methods to identify events from the article collection for a certain topic [1, 30, 27, 9]. They usually work in a two-stage manner. In the first stage, the system identifies important temporal information (year, date, etc.) and assigns them to the articles. In the second stage, a summariser generates a summary for each date. The generated summaries are combined in date order to make up a timeline.

2.2 Preference Learning from Pairwise Labels

To optimise the summarisation model, we need an objective that reflects the quality of the summary in the eyes of the user. Therefore, we map pairwise labels P provided by a human annotator to a score function f using Gaussian Process Preference Learning (GPPL) [26], an extension of the random utility model proposed by Thurstone [28]:

$$p(P(s_1, s_2)|f(s_1), f(s_2)) = \Phi(z), \tag{1}$$

where $P(\cdot)$ is the pairwise preference label indicating that summary s_1 is preferred to s_2, Φ is the cumulative distribution function of the standard normal distribution which is also known as a probit likelihood, and $z = \frac{f(s_1)-f(s_2)}{\sqrt{2\sigma^2}}$. Assuming that the distribution of f is a multivariate Gaussian, the probit $\Phi(z)$ allows us to infer the score function f from pairwise labels by approximate Bayesian inference. The Bayesian approach of GPPL permits inference with small amounts of pairwise labels, thus reducing the cost of the learning process, and accounts for contradictory and incorrect labels [26].

2.3 Reinforcement Learning: Actor-Critic

We use reinforcement learning (RL) to train the abstractive summariser since the preference-based score function is not applicable for supervised learning. For a given reward function R, at the timestep T, the objective L can be written as the expected reward:

$$L(\theta) = E_{\pi_\theta}\left[\sum_{t=0}^{T-1} R_{t+1}\right], \tag{2}$$

where θ is the parameters of the policy function π. The gradient of θ can be written as follows:

$$\nabla L(\theta) = \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)G(t) \tag{3}$$

where $G(t) = \sum_{t'=t+1}^{T} R_{t'}$ is the return starting from the state s_t, and a_t is the action taken at the corresponding timestep.

Actor-Critic introduces a baseline value predicted by the critic $\hat{v}_w$ (parameterised by w) [19]. The advantage of the action a_t is computed to assess how much better it is than taking the average action at the given state s_t.

$$Adv(t) = G(t) - \hat{v}_w(t) \tag{4}$$

Integrating the advantage into the objective function optimises the actor's policy to yield positive returns in the long term, while the critic learns by minimising the mean squared error between the predicted and real values. Therefore, the policy of the actor and the critic enhance each other through the learning process, and the variance of sampling from the policy distribution is less for Actor-Critic comparing to the general Policy Gradient algorithm. Their gradients can be written as follows:

$$\nabla L_{actor}(\theta) = \sum_{t=0}^{T-1} \nabla_\theta log\pi_\theta(a_t|s_t)Adv(t) \tag{5}$$

$$\nabla L_{critic}(w) = \frac{1}{T}\sum_{t=0}^{T-1} \nabla_w \hat{v}_w(s_t) \tag{6}$$

3 Our Method

3.1 Workflow

3.1.1 Baseline

Our method follows the same workflow as CLUST [9], an extractive *event detection* approach. CLUST first encodes source documents into vectors using TF-IDF then clusters them. Each cluster is assigned the date that is mentioned most frequently by the source documents in its article collection. Then the clusters are ranked by the number of times their assigned dates are mentioned throughout the entire set of source documents, and the top-l clusters are selected as the key events. An extractive summariser, CentroidOpt [10], is used by CLUST to select sentences from each key event cluster and concatenate them as a summary.

3.1.2 Contextualised Embedding-based Event Detection

We keep the two-stage event detection approach and change some components to better capture event information. In order to group similar source documents, we use Sentence-BERT [25] to compute

the contextualised embedding of the source documents instead of TF-IDF, as Sentence-BERT embeddings better reflect semantic similarity. We take the average embedding over the sentences as the representation of each document. Prior TLS methods clustered articles using Affinity Propagation (AP) [27] and Markov Clustering (MC) [9]. Therefore, we test these two algorithms as well as Agglomerative Clustering (AC) [20], as these algorithms do not require presetting the number of clusters. The date assignment and cluster ranking mechanism are kept as the same in the baseline (CLUST).

3.1.3 Learning to summarise using PBRL

For the second stage, we integrate PBRL with an abstractive summariser implemented by a pretrained neural network. To generate a summary for each event cluster, the summariser is applied to each cluster in turn. The source documents within each cluster are concatenated to form a single input text. PBRL fine-tunes the summariser leveraging the input text from each cluster. The workflow is shown in Figure 1. The steps are as follows:

1. Before the fine-tuning starts, the user reads and ranks example timelines generated by the extractive baseline and our system using the pretrained abstractive summariser in zero-shot mode. Pairwise preference labels are derived from the ranking.
2. The user provides keywords of interest about the timeline content.
3. The score function f is learned using GPPL mentioned in Section 2.2 and then used to update a sub-reward function as a component to the compound reward.
4. The system visits each cluster in turn and samples hundreds of episodes for each individual cluster. The summariser learns from the episodes by utilising the reward function for reinforcement learning with the Actor-Critic method described in section 2.3.
5. The system generates event summries for each cluster and concatenates them in date order to form the timeline.

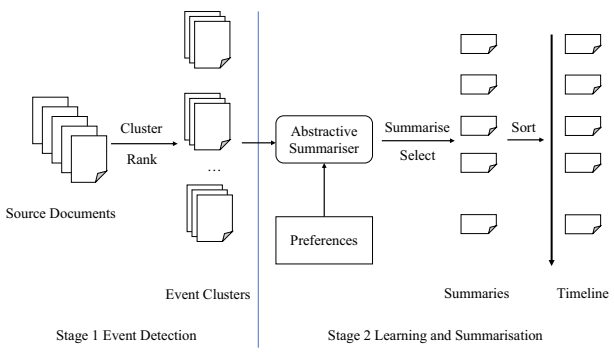

Figure 1. The system works in a two-stage manner. The first stage detects events using a clustering algorithm. The abstractive summariser learns from the preferences and summarises each event. All the generated summaries will be sorted by their date to form a timeline.

3.2 Compound Rewards

Optimising purely for keywords and preference scores could lead to low quality text, as the policy may find shortcuts such as repeating keywords; a score function f learned from few preference labels is insufficient to guard against this. Therefore, some prior RL-based abstractive summarisers [23, 32] combined a supervised learning loss with a reinforcement learning loss converted from a ROUGE-oriented [15] reward. Both losses rely on reference summaries, which may be unavailable and are expensive to create. This motivates us to define several sub-reward functions that ensure complementary qualities of the summary, including adherence to user preferences and interests, consistency with sources, and fluent, non-repetitive text.

3.2.1 Preference Reward

To improve the pretrained summariser, our system acquires two types of preference input: pairwise labels P, which indicate users' high-level preferences over different summary versions, and keywords K, which specify the details they prefer to see in the generated summaries. The coherence between the generated summary s and the user's preferences P, K is assessed through the preference reward R_1, which is a weighted sum of two terms:

$$R_1 = w \sum_{k_i \in K} \cos\langle \vec{k_i}, \vec{s} \rangle + (1-w) f(s), \tag{7}$$

where $\cos\langle \vec{k_i}, \vec{s} \rangle$ is cosine similarity between the embedding of timeline summary s and keyword k_i, and w weights keyword versus pairwise preferences. Both embeddings are computed using sentence-BERT. For the representation of the timeline, we compute the sentence embedding for each event summary, then take the mean. $f(s)$ is learned using GPPL [26] from pairwise preferences P with the average sentence embeddings as the inputs. Learning the score function f only requires about 10 pairwise labels and keywords are easy to provide, hence the cost of preference learning is much lower than that of obtaining reference timelines through crowdsourcing.

3.2.2 Consistency Reward

As well as satisfying user preferences, the generated summary should have high semantic similarity to the source documents to ensure it conveys similar information. We again compute Sentence-BERT embeddings of source articles and event summaries [25] and take the average vectors to represent the input documents and timeline. Prior work [18] has successfully used cosine similarity between embeddings to quantify the consistency of generated text. Thus, we use the same approach to define the consistency reward R_2:

$$R_2 = \cos\langle \vec{s}, \vec{D} \rangle, \tag{8}$$

where $\vec{D}$ is the embedding of the source documents.

3.2.3 Language Quality Reward

Here, we do not have reference summaries to use as teacher forcing input for the decoder, unlike prior work [23]. Without this input, the model may degenerate to outputting topically relevant but non-fluent text. Therefore, we define a language quality reward to maintain the linguistic fluency of the generated output. We take the generated output s as the input of GPT2 [24] and use its loss to evaluate the language quality of the output. The normalisation hyperparameter α is the maximum loss pre-computed on the validation dataset.

$$R_3 = \frac{\alpha - L_{gpt2}(s)}{\alpha} \tag{9}$$

3.2.4 Repetition Penalty

In natural language generation, reinforcement learning agents sometimes trick for high rewards if they learned that generating specific tokens receives high rewards [18]. Therefore, we define R_4 to penalise repetitive generation.

$$R_4 = 1 - \frac{\text{\#repeated\_tokens}}{\text{\#tokens}} \tag{10}$$

3.3 Training

The training process takes a small amount of preferences as input, then reward function R_1 is learned using GPPL over the average sentence embeddings for each timeline, and used as a component of the final reward R when fine-tuning the summariser. The final reward function is computed as the weighted sum of the four sub-rewards. The weights $\gamma_{1,2,3,4}$ assigned to each sub-reward are chosen on the validation dataset.

$$R = \gamma_1 R_1 + \gamma_2 R_2 + \gamma_3 R_3 + \gamma_4 R_4 \tag{11}$$

The final reward R is applied in the Actor-Critic algorithm mentioned in section 2.3, in which the actor (summariser) samples hundreds of trajectories and learns from them. The training process is shown in Algorithm 1.

4 Experiments

4.1 Datasets

We conduct the experiments on three benchmark English TLS datasets, Timeline17 (T17) [31], Crisis [30] and Entities [9]. T17 and Crisis are made up of news articles published by news agencies. The documents in Entities are collected from Wikipedia and the topics are celebrities' biographies. We take 1 timeline from each dataset as the validation set to tune the hyperparameters. Table 2 reveals some of the properties of the three datasets. Crisis and Entities have larger source document pools for each timeline compared to T17, while their reference timelines are more compressed in terms of text and time, which makes it harder to find the right dates for the summaries on the two datasets.

4.2 Model Settings and Evaluation Metrics

In the event detection stage, we use all-MiniLM-L6-v2 [25] to compute the sentence embedding. The distance threshold for Agglomerative Clustering is set to 0.7. To prevent source documents being excessively long, we keep the setting in the baseline, CLUST [9], that takes only the first 5 sentences of each document. We took the pretrained BART-large-xsum [13] as our baseline summariser since it was fine-tuned on a related task (news summarisation). The critic is a linear regression model with input size of 768. We used two AdamW optimisers (learning rates $\alpha_{actor} = 2e{-}4, \alpha_{critic} = 1e{-}3$, $\beta = (0.9, 0.999)$ for both) to optimise the summariser (actor) and the critic in RL. We train our model on an NVIDIA 2080Ti with batch size 1. The number of episodes is 300 for each event cluster.

In terms of preferences, we simulated keyword choices for each timeline by extracting 10 terms with the highest TF-IDF scores from reference timelines, allowing us to compare with the reference summaries in the evaluation. Regarding the pairwise preference labels, we produced five candidate timelines for each topic and hired annotators to rank them. Pairwise labels were then derived from this real human preference ranking.

We evaluated the event detection performance by computing the date overlap between the system timeline and the ground truth. As for the generated content, we conducted the evaluation using BERTScore [34]. It replaces the exact matches in ROUGE by contextualised embeddings, resulting in more robustness as an evaluation metric for generation tasks. However, BERTScore is not able to take the temporal information into consideration, which is greatly related to the quality of a timeline. We therefore applied Alignment-based ROUGE [17], which is specially adapted for TLS, to evaluate the system output from both perspectives. Additionally, we hired human evaluators to assess the legibility, factual consistency, topical coherence, and informativeness of the preferences of the system output, which are tricky for automated metrics to evaluate.

4.3 Results

4.3.1 Event Detection Performance

We first evaluated the system's ability to detect events, measured by its date selection performance. Table 3 indicates that our contextualised embedding-based event detection outperforms the baseline's results on Timeline17 and Crisis but has a small decrease on Entities. We think this is because all the source documents for each timeline in Entities are strongly related to the same topical person. The border between two events in the embedding space is therefore not as clear as in the other two datasets, since the person's name is mentioned repeatedly and the contextualised document embeddings therefore intermingle, leading to clustering errors. In addition, the huge number of candidate dates in the source documents and large number of dates in the reference timelines make it even trickier to find the correct dates. Overall, we find that Agglomerative Clustering performed best in combination with contextualised embeddings, and therefore use AC+CE in our subsequent experiments.

4.3.2 BERTScore

Although ROUGE [15] is the most commonly used evaluation metric for generation tasks, it is also challenged for being less favourable to abstractive summarisation [21] as it can incorrectly penalise paraphrasing because it counts overlapping lexical units in two pieces of text. Therefore, we evaluated our system timeline's content using BERTScore, which uses contextualised embeddings to avoid the need for exact lexical matches. As input to the metric, we concatenated the event summaries in date order for each timeline. Our method outperforms the extractive baseline on BERTScore, and improves over the pretrained summariser without fine-tuning, which shows that abstractive approaches can outperform extractive methods when adapted to TLS.

4.3.3 Adapted ROUGE for TLS

The adapted ROUGE (marked as AR1 and AR2) matches the dates in the generated and reference timeline before computing traditional ROUGE scores. The temporal information also affects the results, because it only computes token overlaps on matched event summaries. Therefore, we use it to comprehensively evaluate the quality of our system's timelines. Results in Table 4 show that our method outperforms the extractive baseline on Timeline17 and Entities, while having a slightly lower but still competitive AR1 result on Crisis.

Since our system outperforms the extractive baseline on both BERTScore and date selection, we believe that paraphrasing in the abstractive summaries may lead to the decrease in AR1 on Crisis.

Table 2. Statistics of three datasets, cited from [9]. All numbers are the average per timeline except the first column. Date compression ratio is the timeline length divided by the number of dates. Sentences compression ratio is the total sentences in each timeline divided by the number of source documents.

Dataset	#Timelines	#Dates	#Docs	Timeline Length	Date Compression Ratio	Sent. Compression Ratio
T17	19	124	508	36	0.43	0.0117
Crisis	22	307	2310	29	0.11	0.0005
Entities	47	600	959	23	0.06	0.0017

Table 3. The date selection F score for different settings. CE refers to contextualised embedding. * indicates that the improvement is significant compared to the baseline (α=0.01).

	T17	Crisis	Entities
MC+TF-IDF (baseline)	0.407	0.226	**0.174**
MC+CE	0.273	0.176	0.160
AP+CE	0.450	0.205	0.129
AC+CE (ours)	**0.490***	**0.239***	0.154

Algorithm 1: Learning the reward function from preferences and fine-tuning the summariser for Timeline T.

Reward learning:
  input : Timelines $\{T_t\}$ used to draw pairwise preferences, the corresponding preference labels over timelines $\left\{P_j^{(T)}\right\}$, wanted keywords $\{k_i\}$
  foreach *timeline* T_t **do**
    $E_t \leftarrow$ `AvgSentenceEmbedding(`T_t`)`;
  $f \leftarrow$ `TrainGPPL(`$\{P_j^{(T)}\}, \{E_t\}$`)`;
  $R_1 \leftarrow$ `UpdateR1(`$f, \{k_i\}$`)`;
  output: Preference reward function, R_1
Summariser fine-tuning:
  input : Event clusters $\{C_i\}$, with corresponding article collection $\{A_i\}$
  foreach *cluster* C_i **do**
    $Source_i \leftarrow$ `Concatenate(`A_i`)`;
    for $j \leftarrow 1$ **to** $max\_episodes$ **do**
      $T \leftarrow$ `SampleSummary(`$Source_i$`)`;
      foreach *token* t_i *of the trajectory* T **do**
        $t_{1\to j} \leftarrow$ tokens in T from 1 to j ;
        $r_j \leftarrow$ `R(`$Source_i, t_{1\to j}$`)`;
        $s_j \leftarrow$ `ComputeStates(`$t_{1\to j}$`)`;
        $\hat{v}_j \leftarrow$ `Critic(`$Source_i, t_{1\to j}$`)`;
        $g_j \leftarrow$ `ComputeReturn(`$s_j, t_{1\to j}$`)`;
        $Adv_j \leftarrow$ `ComputeAdvantage(`$g_j, \hat{v}_j$`)`;
      $Loss_A \leftarrow$ `ComputeActorLoss(`T, Adv`)`;
      $Loss_C \leftarrow$ `ComputeCriticLoss(`$\hat{v}, g$`)`;
      `BackwardPropagation(`$Loss_A, Loss_C$`)`;
  output: Fine-tuned summarisation model
Generation:
  input : Event clusters $\{C_i\}$, with corresponding article collection $\{A_i\}$
  foreach *cluster* C_i **do**
    $s_i \leftarrow$ `GenerateSummary(`$Source_i$`)` ;
  $T \leftarrow$ `SortEventSummaries(`$\{s_i\}$`)`;
  output: Return timeline T generated by the trained summariser for the cluster.

Another probable reason for the lower ROUGE scores on Crisis is the huge number of documents and extremely low sentence compression ratio for each timeline. Condensing massive input into a short but adequate abstractive summary can be an extremely tricky task compared to selecting several sentences near the cluster centroid for the extractive method to cover the main idea.

4.3.4 Human Evaluation

We invited 10 volunteers to assess the timelines from a comprehensive perspective. The participants were requested to mark timelines from 0 to 10 for two randomly-selected topics from T17 and two from Crisis. Each topic had 3 timelines generated by different TLS systems (baseline, ours, ours without PBRL), meaning that each volunteer read and annotated approximately 80 event summaries. Separate scores were given for *legibility*, *informativeness* of the content, *coherence* to the topic, and *factual consistency*. We also encouraged the volunteers to use their own standards to give an *overall* score.

The results in Table 5 demonstrate a major improvement over the extractive baseline in terms of legibility, indicating that our system generates more readable summaries. The improvement in overall scores indicated that our system's output is preferred by the human evaluators. The student t-test between our model and the baseline demonstrates that the improvement in these two aspects is statistically significant. For these two aspects, we computed inter-annotator agreement by mapping the scores to rankings and computing pairwise Cohen's κ. We obtained coefficients of 0.68 for legibility, 0.62 for overall score, and ~0.53 for other all aspects, indicating moderate agreement between users.

4.4 Ablation Study

To better understand the effect of each module in our system, we carried out an ablation study on PBRL and the sub-reward functions. We evaluated the zero-shot setting on the pretrained abstractive summariser, and tested the contribution of the sub-reward functions by setting each one to zero (R3 and R4 we ablated jointly since they both aim to ensure linguistic fluency). We also tested the combination of our clustering method with the extractive summariser used in CLUST, CentroidOpt, to reveal whether the improvements arise purely through the changes we made to event detection.

The results in Table 4 indicate that the abstractive summariser with zero-shot setting can receive competitive results, potentially because the summariser was previously fine-tuned on a similar task. However, PBRL is still needed to surpass the extractive baseline. The results in the next three rows, where each sub-reward is turned off in turn, are all worse than that with the full reward function, demonstrating that human preference feedback is helpful in improving the summariser but must be balanced with topical consistency and, especially, language quality. The bottom row in Table 4 shows that the extractive summariser used in CLUST performs worse than the zero-shot abstractive summariser when combined with our clustering method.

Table 4. The BERTScore and ROUGE score of different system settings. * indicates that the improvement is significant compared to the baseline (α=0.01).

	T17			Crisis			Entities		
	BERT-F	AR1-F	AR2-F	BERT-F	AR1-F	AR2-F	BERT-F	AR1-F	AR2-F
CLUST (baseline)	0.819	0.082	0.020	0.821	**0.061**	**0.013**	0.807	0.051	0.015
Ours	**0.822***	**0.085***	**0.023***	**0.831***	0.059	**0.013**	**0.821***	**0.053***	**0.016**
w/o PBRL	0.816	0.082	0.019	0.823	0.053	0.012	0.811	0.047	0.014
w/o R1	0.820	0.077	0.018	0.829	0.056	0.013	0.817	0.044	0.011
w/o R2	0.817	0.073	0.018	0.830	0.058	0.011	0.818	0.045	0.013
w/o R3+R4	0.811	0.071	0.015	0.825	0.054	0.012	0.815	0.042	0.012
AC+CentroidOpt	0.813	0.078	0.018	0.821	0.056	0.012	0.806	0.043	0.010

Table 5. The human evaluation score for different TLS systems. Higher score means better performance on corresponding aspect.

	Legibility	Informativeness	Topical Coherence	Factual Consistency	Overall
Ext. Baseline	5.8	8.4	**9.2**	8.3	6.7
Ours	**7.9**	**8.8**	8.7	**8.6**	**8.5**
w/o PBRL	6.7	7.7	8.1	7.2	7.6

Considering this result alongside Table 3 shows that the clustering approach needs to be adapted to the chosen summarisation approach.

4.5 Case Study

We conducted a case study to further understand the characteristics of the generated summaries. We selected a topic in Timeline17 and generated three timelines using the extractive baseline (CLUST), our method with and without PBRL. They are displayed with the reference timeline in Table 6. There are web formatting tokens marked in red in the reference timeline (*-LRB-, -RRB-, \u00ac*, etc.), since these summaries were extracted from webpages [31]. These tokens may cause our method's performance to be underestimated, as it does not generate these reference tokens. Excluding these formatting tokens makes the abstractive summaries more legible than the extractive summaries. The blue text highlights irrelevant information such as page update times and the news agency name. The extractive TLS system fails to filter out page update information within the summaries. The two abstractive timeline summaries are much more concise compared to the extractive one as they simply state the event and avoid over-quoting details. However, the abstractive summariser can generate hallucinations (marked in green). The false fact is corrected in the PBRL version, demonstrating that our method may help with the factual consistency of the output, which is concurrent with similar findings for learning from human feedback [22].

5 Related Work

Timeline Summarisation Prior mainstream TLS methods are usually extractive and avoid neural abstractive summarisation because the amount of the data is insufficient for training [6, 9, 14]. Some work built up relatively large datasets to train the neural timeline summariser in a supervised manner [3, 4], but the data is not openly accessible and each timeline is sourced from a single article in an encyclopedia, rather than multiple news sources. While the evaluation used the standard ROUGE that does not consider the date structure of the timeline, the results obtained by Chen et al.[3] show that abstractive summarisation may be able to outperform extractive methods for TLS when adapted correctly to the data. Other work on abstractive TLS managed to obviate model training by utilising the word-graph [27]. Since some prior abstractive TLS methods worked on different datasets or applied their own adapted evaluation metrics, it was not possible to directly compare our results against theirs. To the best of our knowledge, this paper is the first to apply a pretrained summarisation model to TLS and to consider PBRL as a way to align timeline summaries with human preferences, instead of relying on large amounts of training data.

Preference-based Reinforcement Learning While early work on PBRL showed pairwise preferences provide more consistent rewards for RL than absolute scores [7] and that small amounts of human feedback are sufficient to enable offline RL [7, 12]. PBRL has been used to refine single-document summarisation [2], multi-document summarisation [8] and machine translation [12], but these approaches did not use a compound reward function to incorporate keywords and maintain language quality. For conversational agents, a combined reward was shown to be necessary to satisfy competing objectives such as fluency and topicality [18], and has been used to ensure that large language models satisfy user preferences [22]. In summary, the related work does not adopt PBRL to TLS.

6 Conclusion

In this paper, we proposed a novel TLS pipeline. We used contextualised embeddings and applied a more suitable clustering algorithm to better capture the semantic information in the article collection, thus enhancing the system's ability to detect events in news datasets. We leveraged PBRL to fine-tune the pretrained abstractive summariser, improving its performance comparing to the zero-shot setting while minimising the need for training data. Our system outperforms the state-of-the-art clustering-based extractive TLS baseline in terms of BERTScore across all three datasets and achieves better ROUGE scores on two datasets. The human evaluation indicated that our abstractive timelines are more legible than those of the extractive baseline, and that overall, they align more closely with human preferences than summaries produced by the zero-shot abstractive summariser and the extractive baseline.

Table 6. Timeline examples generated by different systems on the involvement of Conrad Murray in the death of Michael Jackson.

Reference	2009-06-25	Dr Murray finds Jackson unconscious in the bedroom of his Los Angeles mansion. Paramedics are called to the house while Dr Murray is performing CPR, according to a recording of the 911 emergency call. He travels with the singer in an ambulance to UCLA medical center where Jackson later dies.
	2009-07-22	The doctor's clinic in Houston is raided by officers from the Drug Enforcement Agency -LRB- DEA -RRB- looking for evidence of manslaughter.
	2010-02-08	Dr Murray is charged with involuntary manslaughter. He pleads not guilty and is released on $75,000 -LRB- \u00ac # 48,000 -RRB- bail. The judge says he can continue to practice medicine, but bans him from administering anesthetic agents, "specifically propofol".
	2011-01-04	Preliminary hearings begin. Prosecutors allege that Dr Murray "hid drugs" before calling paramedics on the day Jackson died. They also state that he did not perform CPR properly and omitted to tell paramedics that he had given Jackson propofol.
	2011-11-03	The case against Dr Murray goes to the jury following closing statements. The prosecution concludes by saying the doctor's care of Jackson had been "bizarre". The defense maintains Dr Murray was not responsible and that the singer caused his own death while his doctor was out of the room. "If it was anybody else , would this doctor be here today?" defense lawyer Ed Chernoff says.
	2011-11-07	Dr Conrad Murray is found guilty of involuntary manslaughter after nine hours of jury deliberations. The doctor was remanded in custody without bail until he receives his sentence.
	2011-11-29	Dr Conrad Murray is sentenced to four years in county jail. Judge Michael Pastor says the evidence in the case showed him guilty of a "continuous pattern of lies and deceit".
Extractive Baseline	2009-06-25	Dr Conrad Murray is on trial accused of the involuntary manslaughter of singer Michael Jackson. 29 September 2011 Last updated at 15:44 GMT Help Live coverage of the trial of Michael Jackson's personal physician, Dr Conrad Murray, who is charged with involuntary manslaughter of the singer.
	2009-07-22	Michael Flanagan of the DEA describes the operation Police have searched the Las Vegas home and offices of Michael Jackson's doctor as part of a manslaughter investigation into the singer's death. Dr Conrad Murray, who police say is not a suspect, was at Jackson's mansion and tried to revive him before he died.
	2010-02-08	Dr Murray has denied he caused Michael Jackson's death Los Angeles prosecutors will file a criminal case against Michael Jackson's doctor on Monday in connection with the singer's death, officials say.
	2011-01-06	Prosecutors say his negligence led to Jackson's death aged 50 in June 2009. Michael Jackson doctor Conrad Murray 'hid drug dose' Dr Murray, shown here in February, joined Jackson's entourage to get him fit for London concerts Michael Jackson's doctor tried to hide the fact that he had given the singer a powerful anesthetic shortly before he died, a prosecutor has said.
	2011-11-03	Conrad Murray decides not to give evidence Defense lawyers for Dr Conrad Murray say they will call no further witnesses Dr Conrad Murray has told a judge he will not testify in his own defense in the trial over Michael Jackson's death. Dr Murray told trial Judge Michael Pastor he had made his decision "freely and explicitly", reports said.
	2011-11-07	Page last updated at 08:14 GMT, Tuesday, 8 November 2011 Propofol is a fast-acting hospital sedative used before anaesthetics Michael Jackson's physician, Doctor Conrad Murray, has been found guilty of involuntary manslaughter by a jury in Los Angeles. 8 November 2011 Last updated at 06:28 Michael Jackson's doctor, Conrad Murray, found guilty Michael Jackson 's doctor has been found guilty of causing the singer 's death.
	2011-11-29	The trial of Michael Jackson's doctor, Conrad Murray, took place on the ninth floor of the superior court of Los Angeles. The Atlantic Murray will be sentenced on 29 November, and was held without bail after his conviction The last doctor of Michael Jackson's life was convicted of manslaughter by a California jury because the law demanded it and the facts supported it.
Ours w/o PBRL	2009-06-25	The personal doctor of Michael Jackson has pleaded not guilty to involuntary manslaughter in connection with the singer's death.
	2009-07-22	The US Drug Enforcement Administration (DEA) has searched the Los Angeles clinic of Michael Jackson's doctor.
	2010-02-08	Dr Conrad Murray, the man accused of killing Michael Jackson, has returned to work at a Las Vegas hospital.
	2011-01-06	Michael Jackson's doctor Conrad Murray ordered a bodyguard to pick up vials of medicine before phoning for help on the day he died, his trial has heard.
	2011-11-03	The jury in the involuntary manslaughter trial of a doctor accused of giving Michael Jackson a fatal dose of propofol has retired to consider its verdict.
	2011-11-07	The trial of Michael Jackson's doctor took place on the ninth floor of a Los Angeles court.
	2011-11-29	The judge in the case of Michael Jackson's former doctor Conrad Murray has said he will not sentence him to prison under a new California law.
Ours	2009-06-25	The personal doctor of Michael Jackson has been ordered to stand trial for involuntary manslaughter over the singer's death of a drug overdose.
	2009-07-22	The Drug Enforcement Agency (DEA) has obtained a warrant to search the home of Michael Jackson's doctor, Conrad Murray.
	2010-02-08	The former doctor of Michael Jackson has pleaded not guilty to the singer's manslaughter.
	2011-01-06	Michael Jackson's doctor ordered a bodyguard to pick up vials of medicine before phoning for help, a court in Los Angeles has heard.
	2011-11-03	The jury in the trial of a Los Angeles doctor accused of involuntary manslaughter over Michael Jackson's death has been sent home for the weekend.
	2011-11-07	The trial of Michael Jackson's doctor has been taking place in Los Angeles.
	2011-11-29	Michael Jackson's former doctor Conrad Murray has been sentenced to four years in prison for involuntary manslaughter.

Acknowledgements

We thank the reviewers for their valuable comments. This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol - http://www.bristol.ac.uk/acrc/. The financial support for Yuxuan Ye was provided by the programme of the China Scholarship Council (No. 202108060154).

References

[1] Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen, 'Predicting relevant news events for timeline summaries', in *Proceedings of the 22nd International Conference on World Wide Web*, pp. 91–92, (2013).

[2] Florian Böhm, Yang Gao, Christian M Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych, 'Better rewards yield better summaries: Learning to summarise without references', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3110–3120, (2019).

[3] Xiuying Chen, Zhangming Chan, Shen Gao, Meng-Hsuan Yu, Dongyan Zhao, and Rui Yan, 'Learning towards abstractive timeline summarization.', in *IJCAI*, pp. 4939–4945, (2019).

[4] Xiuying Chen, Mingzhe Li, Shen Gao, Zhangming Chan, Dongyan Zhao, Xin Gao, Xiangliang Zhang, and Rui Yan, 'Follow the timeline! generating an abstractive and extractive timeline summary in chronological order', *ACM Transactions on Information Systems*, **41**(1), 1–30, (2023).

[5] Weiwei Cheng, Johannes Fürnkranz, Eyke Hüllermeier, and Sang-Hyeun Park, 'Preference-based policy iteration: Leveraging preference learning for reinforcement learning', in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Part I 11*, pp. 312–327. Springer, (2011).

[6] Hai Leong Chieu and Yoong Keok Lee, 'Query based event extraction along a timeline', in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 425–432, (2004).

[7] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei, 'Deep reinforcement learning from human preferences', in *Advances in Neural Information Processing Systems*, eds., I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, volume 30. Curran Associates, Inc., (2017).

[8] Yang Gao, Christian M Meyer, and Iryna Gurevych, 'Preference-based interactive multi-document summarisation', *Information Retrieval Journal*, **23**, 555–585, (2020).

[9] Demian Gholipour Ghalandari and Georgiana Ifrim, 'Examining the state-of-the-art in news timeline summarization', in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1322–1334, (2020).

[10] Demian Gholipour Ghalandari, 'Revisiting the centroid-based method: A strong baseline for multi-document summarization', in *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 85–90. Association for Computational Linguistics, (2017).

[11] David C Kingsley and Thomas C Brown, 'Preference uncertainty, preference learning, and paired comparison experiments', *Land Economics*, **86**(3), 530–544, (2010).

[12] Julia Kreutzer, Joshua Uyheng, and Stefan Riezler, 'Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning', in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1777–1788, (2018).

[13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, 'Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension', in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, (2020).

[14] Manling Li, Tengfei Ma, Mo Yu, Lingfei Wu, Tian Gao, Heng Ji, and Kathleen McKeown, 'Timeline summarization based on event graph compression via time-aware optimal transport', in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6443–6456. Association for Computational Linguistics, (2021).

[15] Chin-Yew Lin, 'Rouge: A package for automatic evaluation of summaries', in *Text summarization branches out*, pp. 74–81, (2004).

[16] Hui Lin and Jeff Bilmes, 'A class of submodular functions for document summarization', in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 510–520. Association for Computational Linguistics, (2011).

[17] Sebastian Martschat and Katja Markert, 'A temporally sensitive submodularity framework for timeline summarization', in *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 230–240, (2018).

[18] Mohsen Mesgar, Edwin Simpson, and Iryna Gurevych, '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*, pp. 549–562, (2021).

[19] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu, 'Asynchronous methods for deep reinforcement learning', in *International conference on machine learning*, pp. 1928–1937. PMLR, (2016).

[20] Fionn Murtagh and Pedro Contreras, 'Algorithms for hierarchical clustering: an overview', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **2**(1), 86–97, (2012).

[21] Jun Ping Ng and Viktoria Abrecht, 'Better summarization evaluation with word embeddings for rouge', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1925–1930, (2015).

[22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al., 'Training language models to follow instructions with human feedback', *Advances in Neural Information Processing Systems*, **35**, 27730–27744, (2022).

[23] Romain Paulus, Caiming Xiong, and Richard Socher, 'A deep reinforced model for abstractive summarization', in *International Conference on Learning Representations*, (2018).

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., 'Language models are unsupervised multitask learners', *OpenAI blog*, **1**(8), 9, (2019).

[25] Nils Reimers and Iryna Gurevych, 'Sentence-bert: Sentence embeddings using siamese bert-networks', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, (11 2019).

[26] Edwin Simpson and Iryna Gurevych, 'Scalable Bayesian preference learning for crowds', *Machine Learning*, **109**(4), 689–718, (2020).

[27] Julius Steen and Katja Markert, 'Abstractive timeline summarization', in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 21–31. Association for Computational Linguistics, (2019).

[28] Louis L Thurstone, 'A law of comparative judgment.', *Psychological review*, **34**(4), 273, (1927).

[29] Giang Tran, Mohammad Alrifai, and Eelco Herder, 'Timeline summarization from relevant headlines', in *European Conference on Information Retrieval*, pp. 245–256. Springer, (2015).

[30] Giang Binh Tran, Mohammad Alrifai, and Eelco Herder, 'Timeline summarization from relevant headlines.', in *ECIR*, volume 9022, pp. 245–256, (2015).

[31] Giang Binh Tran, Tuan A Tran, Nam-Khanh Tran, Mohammad Alrifai, and Nattiya Kanhabua, 'Leveraging learning to rank in an optimization framework for timeline summarization', in *SIGIR 2013 Workshop on Time-aware Information Access (TAIA*, (2013).

[32] Shweta Yadav, Deepak Gupta, Asma Ben Abacha, and Dina Demner-Fushman, 'Reinforcement learning for abstractive question summarization with question-aware semantic rewards', in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 249–255, (2021).

[33] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu, 'Pegasus: Pre-training with extracted gap-sentences for abstractive summarization', in *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, (2020).

[34] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi, 'Bertscore: Evaluating text generation with bert', in *International Conference on Learning Representations*, (2019).

[35] Markus Zopf, 'Estimating summary quality with pairwise preferences', in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1687–1696, New Orleans, Louisiana, (June 2018). Association for Computational Linguistics.