Placing (Historical) Events on a Timeline: A Classification cum Co-ref Resolution Approach

Anonymous ACL submission

Abstract

The event timeline provides one of the most effective ways to visualize the important historical events that occurred over a period of time, presenting the insights that may not be so apparent from reading the equivalent information in textual form. By leveraging generative adversarial learning for important event classification and by assimilating knowledge based tags for improving the performance of event coreference resolution we introduce a two staged system for event timeline generation from multiple (historical) text documents. In addition, we propose a *vis-timeline* based visualization technique to portray the event timeline. We demonstrate our results on two very well known historical documents – the *Collected Works of Mahatma Gandhi* (CWMG) and the *Collected Works of Abraham Lincoln* (CWAL). Our results can be extremely helpful for historians, in advancing research in history and in understanding the socio-political landscape of a country as reflected in the writings of political leaders/scholars. Our work has some parallels with timeline summarization (TLS) tasks and therefore we use these as baselines. Rigorous experiments demonstrate *that prior event detection which was hitherto absent in the TLS methods* can improve summarization performance. In order to show that our methods are very generic we reuse our method to visualize the evolution of coronavirus related events in India from a collection of various COVID-19 articles. We plan to release the annotated dataset upon acceptance.

1 Introduction

Timeline serves as one of the most effective and easiest means to contextualize and visualize a complex situation ranging from grasping spatio-temporal events in historical studies to critical decision making in businesses. With the stupendous increase of textual resources for many historical contents in several online platforms it has become imperative for the history researchers to understand the chronological orderings of the incessant historical phenomenon. The event timeline can be an extremely useful aid to highlight the temporal and causal relationships among several events and the interactions of the characters over time, that results in identifying common themes that arise over the period of interest in a historical document. For instance, the following timeline in Figure 1 can be remarkably helpful to recognize the context and the actors of a particular event in a certain period.

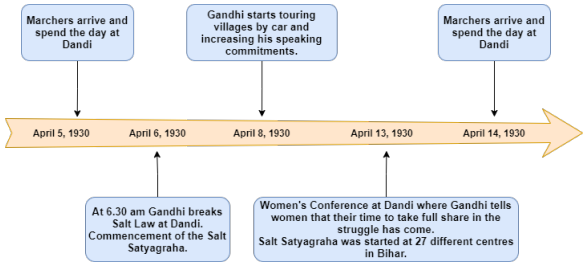

Figure 1: Sample event timeline example extracted from documents.

In this paper we present a full pipeline to build a chronology of events extracted from historical text.

Our contributions are as follows.

- We prepare two datasets one taken from the *Collected Works of Mahatma Gandhi* (CWMG) and the other taken from *Collected Works of Abraham Lincoln* (CWAL) for our experiments. We suitably annotate the datasets for the different follow up tasks enumerated below.
- We first classify important events from the historical text at different instants of time. For this we have manually annotated important sentences (see section 4.2) from a set of the articles chosen from each of the mentioned datasets and trained a generative adversarial learning based classifier (three classes – *event/fact*, *demand*, *other*) to achieve a final macro F1-score of 0.69 for CWMG and 0.65 for CWAL.

- Once the important sentences are classified we perform coreference resolution to merge sentences corresponding to the same event. We use an unsupervised clustering technique to achieve this and obtain an average F1-score of 0.55 for CWMG and 0.47 for CWAL. We further introduce a novel strategy including the temporal information from the extracted sentences and tags generated from world knowledge and pump these into the model to obtain a huge boost in average F1-score (0.68 and 0.63 for CWMG and CWAL respectively). As a follow-up we also developed a supervised deep neural architecture for the coreference resolution task by introducing the novel concept of *event mention-pairs*. Consequently, the macro F1-score of CWMG and CWAL increases to 0.72 and 0.64 respectively.
- In order to establish the generalizability of our approach we attempt to extend the coreference resolution task to a completely orthogonal dataset – COVID-19 events in India. We show that both our unsupervised and supervised approaches perform very well for this dataset. In fact, for the supervised model we achieve a very high F1-score of 0.94 for this task.
- Our method has some parallels with timeline summarization (TLS) tasks. We therefore compare it with the existing state-of-the-art TLS methods on several benchmark TLS datasets as well as on our datasets. One of the very important observation is that prior event detection which has so far not been explored in the TLS literature can have a significant impact on the summarization performance especially in the context of historical corpora.
- Finally, we present an elegant visualisation of the obtained results for easy readability and interpretation. In order to determine the readability and usefulness in the timeline, we conducted an online crowd-sourced survey. Overall, 79% participants found it easily readable and 93% participants found it to be effective in summarizing historical timeline of events.

2 Related work

Important event classification: Zhang and Wallace (2016) used CNN to analyse sensitivity for text classification. Miyato et al. (2017) and Zhang et al. (2020) introduced virtual adversarial training methods for robust text classification from a small number of training data points.

Event coreference resolution: Recent works like Choubey and Huang (2017), Kenyon-Dean et al. (2018) have used neural network based architecture to train their model on benchmark coreference dataset (ECB+ Cybulska and Vossen (2014)). Lu et al. (2020) attempted to create an end-to-end event coreference resolution system based on the standard KBP dataset[1].

Timeline of historical events: Bamman and Smith (2014) proposed an unsupervised generative model to construct the timeline of biographical life-events leveraging encyclopaedic resources such as Wikipedia. Aprosio and Tonelli (2015) also uses Wikipedia for timeline construction of historical events. Bedi et al. (2017) attempted to construct an event timeline from history textbooks considering the sentences having temporal expressions. Adak et al. (2020) created an AI-enabled web portal by digitizing the textual resources from the Collected Works of Mahatma Gandhi (Preservation and Trust, 2013).

Timeline summarization (TLS): The timeline summarization task aims to summarize time evolving documents, which models the input documents along with their temporal information unlike traditional document summarization. Gholipour Ghalandari and Ifrim (2020) evaluated existing state-of-the-art methods for news timeline summarization and proposed *datewise* and *clustering* based approaches on the TLS datasets. Born et al. (2020) demonstrated the potential of employing several IR methods on TLS tasks based on a large news dataset. La Quatra et al. (2021) proposes a new approach by generating date level summaries, and then selecting the most relevant dates for the timeline summarization.

The present work: This work is in line with the event timeline summarization (TLS) task but on a general historical corpora. Previous TLS researchers mostly worked on the documents containing multiple news events, which are rich in events. These works have not focused much on prior event detection and have not addressed how they can be effectively generalized in historical text documents such as biographies. In this work we propose for the first time a novel two-step approach for event timeline generation. To this end, we first

[1] https://www.ldc.upenn.edu/collaborations/past-projects/tac-kbp

adapt GAN-BERT (Croce et al., 2020), a generative adversarial learning framework built on top of the BERT (Devlin et al., 2019) architecture for important event identification from historical text documents. Next, we propose a novel tag curation-cum-embedding technique from world knowledge in order to significantly improve the performance of the unsupervised event coreference resolution methods. As a natural follow-up we also develop a supervised *event mention-pair* based deep neural model for the event coreference resolution task. We compare the proposed method with various TLS baselines and report superior performance.

3 Model architecture

Figure 2 shows the overall architecture of the system. It consists of three major components: (i) important sentence extraction, (ii) event coreference resolution, and (iii) timeline visualization. The arrows represent the direction of data flow. Next, we describe the requirement of each of these components.

Important sentence extraction: This module expects raw English text documents with publication date as input. The publication date serves as the initial reference date for all the sentences in a document. Text pre-processing and sentence extraction is done in this phase. Inspired by Chakraborty et al. (2020), we have used active learning to generate more annotated examples for the minority class to reduce class imbalance. After that the sentences are passed through a classifier to predict important sentences.

Event coreference resolution: As the output of the sentence classification phase we have a set of important sentences from the documents. Many of these sentences may refer to the same event. Therefore we carried out the event coreference resolution to merge the sentences which belong to the same event. To further improve the performance we extracted and used (i) temporal expression from sentences (if any) and (ii) world-knowledge based tags.

Timeline visualization: Once the event coreference resolution phase was successfully executed, we generated visualization for the given event sequence using *vis-timeline*[2], a dynamic, browser based visualization library.

[2] https://visjs.github.io/vis-timeline/docs/timeline/

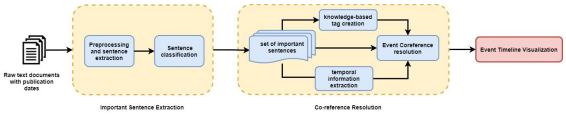

Figure 2: The overall architecture for generating the event timeline.

4 Data preparation

In this section we present the details of the datasets that we prepare for our experiments. We also outline the overall annotation process of these datasets.

4.1 Datasets

Collected works of Mahatma Gandhi: We leverage the Collected Works of Mahatma Gandhi (CWMG) available at (Preservation and Trust, 2013), an assortment of 100 volumes consisting of the books, letters, telegrams written by Mahatma Gandhi and also the compiled writings of the speeches, interviews engaging Gandhi. This data covers many important historical events within the time period of 1884-1948 in British colonised India.

Collected works of Abraham Lincoln: The second dataset we have use to demonstrate our system is based on the life-long writings of the 16th president of the United States, Abraham Lincoln, formally known as the Collected Works of Abraham Lincoln (CWAL)[3] comprising a total of 8 volumes.

COVID-19 event dataset: In addition, to establish the generalizability of the approach, we collect 140 major events, that happened in India during the COVID-19 pandemic from different sources such as *Wikipedia*[4], *Who.int*[5] to be placed on a timeline for elegant visualisation using our system.

4.2 Annotation

In this section we outline the data annotation procedure for the two phases. Note that while the event classification phase is supervised (Level I annotations), the coreference resolution is done using both unsupervised and supervised techniques. The annotations for the coreference resolution (Level II annotations) are therefore required to (a) train the supervised approach and (b) test the efficacy of both the unsupervised and supervised approach.

[3] https://quod.lib.umich.edu/l/lincoln/

[4] https://en.wikipedia.org/wiki/COVID-19_pandemic_in_India

[5] https://www.who.int/india/emergencies/coronavirus-disease-(covid-19)/india-situation-report

To ameliorate the reliability of the annotators, we carried out a trial round of annotations for 100 sentences.

Level I – Important sentences: From the 100 volumes of text files from CWMG we first extract all the letters containing the publication dates and recipients name. There were a total of 28531 letters in the entire CWMG. We primarily use the letters for our experiments as we observe that they contain the best temporal account of the events.

From the overall set of letters, we select the year range 1930–1935 since this range has the largest collection of letters In order to further choose the right data sample, we categorize the letters into *formal* and *informal* types based on the recipients of the letters. A simple heuristic that we follow is – the letters written to government officials and famous historic personalities can be categorized as formal while those written to the family members can be classified as informal ones. We collect the list of Mahatma Gandhi's family member names from Gandhian experts for identifying the informal letters. We manually notice that the formal letters contained much more useful historic information than the informal ones. We therefore only consider the formal letters for manually annotating the useful sentences. In addition, we only consider the letters which have more than 1000 words in its content. This results in 41 letters with substantial content.

Finally, out of these filtered letters we manually annotate all the sentences of 18 letters (i.e., 979 sentences in all). The remaining sentences (i.e., 1689 in total) from the rest of the letters were left unlabelled. Both of these labelled and unlabelled sentences were used for training the classifier. The classes in which the sentences were classified were based on their historical importance. In specific, we identify three such important classes – (a) the *events/fact*, which typically represent that something happened or took place (Pustejovsky et al., 2003). This may consist of participants and locations; (b) the *demands*, which represent the demands Mahatma Gandhi had made to the British government throughout his writings and (c) others. In order to further enrich the dataset we collect gold standard events related to Mahatma Gandhi from an additional reliable and well maintained resource[6].

For the CWAL we simply extract all the sentences from volume 2 and follow similar approaches to annotate important sentences as in the case of CWMG. Without considering any filtering criteria we consider all the 111 articles of volume 2 including his letters and propositions which consist of a total of 1386 sentences.

For both the datasets three annotators annotated the sentences. The inter-annotator agreements, i.e., Cohen's κ were 0.66 and 0.58 for the former and the latter datasets respectively. Table 1 shows the category distribution for both the datasets.

Classes	Count	
	CWMG	CWAL
event/fact	716	200
demand	81	96
other	268	382

Table 1: Category distribution for the two datasets.

Level II – Coreference resolution: The second round of annotation was carried out for evaluating the event coreference detection task on the same dataset. For this case we only annotate the texts which were marked important during the Level-I annotation procedure. In addition to it the Level-II annotation was also carried out for the COVID-19 event dataset. Based on the perception of the annotators, the sentences which potentially referred to the same event were placed in the same cluster. In this case, the inter-annotator agreements were 0.74, 0.61, and 0.78 for the CWMG, the CWAL and the COVID-19 event dataset respectively. In this case, for measuring the annotator agreement we use the MUC (Vilain et al., 1995) based F1-score (Ghaddar and Langlais, 2016).

5 Details of the individual modules

In this section we describe in detail the methods used for important event extraction and coreference resolution.

5.1 Important event extraction

Baselines: As baselines, we use *SVM* (Hearst, 1998) and *Multinomial Naïve Bayes* (Kibriya et al., 2004) on simple bag-of-words feature. For *SVM* we use linear kernel. For the evaluation of the classifiers we use a 70:30 train-test split of the annotated data.

Fine-tuned BERT: Apart from the above two baselines, we try BERT (Devlin et al., 2019) neural network based framework for the classification. We train the model using the PyTorch (Paszke et al.,

[6] https://www.gandhiheritageportal.org/

2019) library, and apply *bert-base-uncased* pre-trained model for text encoding. We use a batch size of 32, sequence length of 80 and learning rate of $2e-5$ as the optimal hyper-parameters for training the model.

GAN-BERT text classifier: In search for further enhancement of the performance based on our limited sets of labelled data, we employ the *GAN-BERT* (Croce et al., 2020) deep learning framework for classifying the important sentences. It uses generative adversarial learning to generate augmented labelled data for semi-supervised training of the transformer based BERT model. It improves the performance of BERT when training data is scarce and is therefore highly suited for our case. Here we also feed the unlabeled data sample, as discussed in section 4.2, to help the network to generalize the representation of input texts for the final classification (Croce et al., 2020).

5.2 Event coreference resolution

Once the classification was done we end up with 'eventful' sentences linked to its corresponding document publication time in the format noted in Table 2.

Doc publication time	Important sentences
03/05/1930	He was arrested at 12.45 a.m. on May 5.
03/05/1930	In Karachi, Peshawar and Madras the firing would appear to have been unprovoked and unnecessary.

Table 2: Sample list of sentences from CWMG after the sentence classification.

Time within sentences: For generating the accurate event timeline we need to assign a valid date to a particular sentence (or event). For example, in the first sentence in Table 2, although the document publication time is mentioned to be 03/05/1930, the sentence clearly has embedded in it the exact event date 05/05/1930 apparent from "arrested on May 5". Therefore, wherever available, we also consider the explicit mention of time inside the sentence to get the exact occurrence time of an event. We extract the explicit mention of time using the *HeidelTime*[7] tool.

Tag generation from world knowledge: An individual sentence does not always contain much information about the event which it is getting referred to. So we attempt to incorporate world knowledge for each individual sentence. By using each sentence as a query we gather the top five *Google* search results using the *googlsearch* api[8] and also consider the document from which the sentence was being extracted. Next we analyse the search result using *TextRank*[9], *Rake*[10] and *pointwise mutual information*[11] to generate top keywords present in the search result. Although these methods produce reasonably good results, in many cases we needed to manually filter out certain noisy tags. For each sentence we therefore land up with one or more tags. We retain the top ten tags for every sentence which means that the number of tags for a sentence could vary between one and ten. We do not use encyclopaedic resources such as Wikipedia to get the search results because the datasets we are using, are only available in a few very specific websites. The pre-trained *sentence-bert* embedding technique was used for obtaining a 768 dimensional representation of the keywords.

Event clustering: We employ several unsupervised approaches for sentence coreference resolution. As baselines, we choose two commonly used approaches for coreference resolution – (a) *Lemma*: It attempts to put the sentence pairs in same coreference chain which share the same head lemma, (b) *Lemma-δ*: In addition to same head lemma as a feature, it also computes the cosine similarity (δ) between the sentence pair based on *tf-idf* features, and only places the sentence pairs in the same coreference chain if δ exceeds some threshold. Then the sentence clusters were created using agglomerative clustering method. To extract the head lemma of a sentence, we use the *SpaCy* dependency parser.

Apart from these two common baselines, we vectorize the sentences using *tf-idf* vectorization technique and then apply different clustering techniques such as *Gaussian-Mixture*[12] model, *agglomerative clustering* to cluster the sentences corresponding to similar events.

We also use the pre-trained *sentence-bert* (Reimers and Gurevych, 2019) model to encode the sentences and apply similar clustering techniques.

Finally, we concatenate the sentence embedding

[7] https://github.com/HeidelTime/heideltime

[8] https://github.com/MarioVilas/googlesearch

[9] https://github.com/DerwenAI/pytextrank

[10] https://pypi.org/project/rake-nltk/

[11] https://www.nltk.org/howto/collocations.html

[12] https://scikit-learn.org/stable/modules/mixture.html

with the tag embedding generated from that particular sentence. We again cluster the sentences based on this new representation. This, as we shall later see, significantly improves the performance of the clustering phase.

We evaluate the clustering results on the basis of the annotated data which had been obtained in the second phase of data annotation.

Supervised event mention-pair model: An *event mention* is a sentence or phrase that defines an event and one event may contain multiple *event mentions* (Chen et al., 2009). Based on our annotated dataset we adopt a supervised two-step *mention-pair* model for the coreference resolution. In the first step we train a binary classifier to determine whether two event mentions are coreferent. After training, the resulting mention-pair model can be applied to classify the test instances. In the second step we employ *agglomerative clustering* to coordinate the pairwise decisions and construct a partition.

We first create a dataset containing all the possible pairs of *eventful* (i.e., event/fact or demand) sentences from the ground truth annotations. We set the coreference label to 1 if the sentence pair is contained in the same cluster as per the Level-II annotation and 0 otherwise. Here we again use a 70:30 split to generate training and test instances. The overall architecture is inspired from Barhom et al. (2019) (see Appendix A.2). The inputs to the model are the two sentences (i.e. S_1 and S_2) and their corresponding *actions* (i.e., A_1 and A_2), *time* (i.e., T_1 and T_2) and *tags* (i.e., K_1 and K_2). We extract *actions* (i.e., A_i) for each of the sentences (fact or demand might not contain any *action*) using *SpaCy* dependency parser.

We encode each feature using pre-trained *GloVe* (Pennington et al., 2014) embedding (100 dimension). Each sentence embedding and its corresponding feature embeddings are then passed through LSTM (Hochreiter and Schmidhuber, 1997) layers and concatenated to generate a mention representation. Two mention representations are finally concatenated to get a pairwise representation and passed through a feed forward network to return a score denoting the likelihood that two mentions are coreferent. Based on the predicted pairwise score on the test instances we use a threshold (0.5) to generate a similarity matrix of the mentions, and then apply agglomerative clustering to partition the similar mentions into different clusters.

6 Experiments

6.1 Evaluation metrics

We have used separate evaluation metrics for the two phases.

Important sentence classification: In this case we use the standard *accuracy* and *F1-score* values.

Dataset	Model	Evaluation Metric	
		Accuracy	F1
CWMG	MNB	0.74	0.45
	SVM	0.79	0.5
	Fine-tuned BERT	0.8	0.57
	GAN-BERT	**0.9**	**0.69**
CWAL	MNB	0.6	0.3
	SVM	0.6	0.34
	Fine-tuned BERT	0.61	0.56
	GAN-BERT	**0.7**	**0.65**

Table 3: Results (accuracy and macro F1-score) for the important event classification using our approaches on the two datasets. MNB: Multinomial Naïve Bayes. Best results are marked in boldface and highlighted in green cells.

Event coreference resolution: Here we conduct the evaluation based on the widely used coreference resolution metrics – (a) *MUC* (Vilain et al., 1995), a link-based metric; (b) B^3 (Bagga and Baldwin, 2000), a mention based metric; (c) *CEAF* (Luo, 2005) which uses a similarity measure (ϕ) to evaluate the similarity of two entities. It uses the Kuhn-Munkres algorithm (Kuhn, 1955) to find the best one-to-one mapping of the key to the response entities using the given similarity measure; and (d) *BLANC* (Recasens and Hovy, 2011), a link-based metric that adapts the Rand Index (Rand, 1971) to coreference resolution evaluation.

Due to the inconsistency of each of these evaluation metrics (Moosavi and Strube, 2016) we shall also report the average outcomes of all the metrics.

6.2 Results

We evaluate the two different phases separately. Ground-truth data was used from each phase for respective evaluations.

Important event classification: The key results for the two datasets (CWMG and CWAL) are summarised in Table 3. Our approach based on GAN-BERT by far outperforms the standard baselines. For the CWMG dataset, the macro F1-score shoots from 0.50 (SVM) to 0.69 on the three class classification task. Likewise for the CWAL dataset, the macro F1-score shoots from 0.34 (Naïve Bayes) to 0.65.

Evaluation of coreference resolution: For the evaluation of event coreference resolution we use several coreference resolution metrics to analyse the model performance. It is apparent from Table 4 that the ap-

Dataset	System	MUC	B^3	CEAF_E	BLANC	Avg (overall)		
		F1	F1	F1	F1	Recall	Precision	F1
CWMG	Lemma	0.45	0.38	0.20	0.49	0.39	0.38	0.38
	Lemma-δ	0.53	0.41	0.19	0.48	0.48	0.40	0.41
	tf-idf + GM	0.53	0.53	0.36	0.60	0.49	0.52	0.50
	tf-idf + AC	0.55	0.50	0.42	0.57	0.50	0.53	0.51
	s-bert + GM	0.61	0.54	0.41	0.60	0.54	0.54	0.54
	s-bert + AC	0.63	0.57	0.40	0.61	0.55	0.56	0.55
	+ tag embedding							
	tf-idf + GM	0.64	0.57	0.45	0.64	0.57	0.60	0.58
	tf-idf + AC	0.62	0.61	0.51	0.66	0.58	0.63	0.60
	s-bert + GM	0.65	0.62	0.48	0.66	0.60	0.60	0.60
	s-bert + AC	0.75	**0.70**	0.52	**0.73**	0.65	**0.71**	0.68
	mention-pair model	**0.91**	0.59	**0.83**	0.53	**0.83**	0.69	**0.72**
CWAL	Lemma	0.28	0.11	0.17	0.49	0.26	0.27	0.27
	Lemma-δ	0.31	0.15	0.14	0.48	0.28	0.27	0.18
	tf-idf + GM	0.53	0.37	0.35	0.49	0.42	0.45	0.43
	tf-idf + AC	0.57	0.42	0.38	0.49	0.45	0.49	0.46
	s-bert + GM	0.43	0.39	0.40	0.54	0.43	0.46	0.44
	s-bert + AC	0.51	0.42	0.40	0.54	0.46	0.48	0.47
	+ tag embedding							
	tf-idf + GM	0.74	0.52	0.40	0.63	0.56	0.59	0.57
	tf-idf + AC	0.72	0.51	0.48	0.64	0.57	0.61	0.59
	S-bert+ GM	0.74	0.41	0.34	0.67	0.51	0.57	0.54
	s-bert + AC	0.82	**0.53**	0.44	**0.72**	0.60	**0.66**	0.63
	mention-pair model	**0.96**	0.42	**0.78**	0.35	**0.82**	0.65	**0.64**
COVID-19	Lemma	0.55	0.39	0.28	0.55	0.51	0.42	0.44
	Lemma-δ	0.34	0.29	0.25	0.51	0.35	0.34	0.35
	tf-idf + GM	0.56	0.41	0.36	0.60	0.47	0.50	0.48
	tf-idf + AC	0.59	0.45	0.36	0.62	0.49	0.54	0.51
	s-bert + GM	0.63	0.45	0.32	0.57	0.47	0.51	0.49
	s-bert + AC	0.61	0.44	0.35	0.57	0.48	0.50	0.49
	+ tag embedding							
	tf-idf + GM	0.44	0.33	0.28	0.54	0.39	0.40	0.39
	tf-idf + AC	0.44	0.34	0.32	0.44	0.4	0.42	0.41
	s-bert + GM	0.57	0.41	0.35	0.59	0.47	0.49	0.48
	s-bert + AC	0.63	0.46	0.39	0.59	0.51	0.52	0.52
	mention-pair model	**0.95**	**0.94**	**0.93**	**0.94**	**0.943**	**0.942**	**0.94**

Table 4: Event coreference results before and after tag embedding. GM: Gaussian Mixture based clustering; AC: Agglomerative Clustering; s-bert: sentence-bert. Best results including the tag embedding are marked in boldface and highlighted in green cells. Best results excluding the tag embedding are marked by underline and highlighted in blue cells.

proach based on clustering with *sentence-bert* embeddings by far outperforms the baselines *lemma* and *lemma-δ*. For the CWMG dataset, *sentence-bert* + agglomerative clustering is the best overall; for the other two datasets no single method is a clear winner.

However, the primary point that we wish to emphasize in the table is the result after incorporating tag embedding. It can be clearly observed that this intuitive, albeit hitherto unreported, technique almost always produces better results. In fact, the assimilation of the tag embeddings with the *sentence-bert* embeddings boosted the overall F1-score by 13%, and 16% for the CWMG and the CWAL datasets respectively. An interesting observation is that the benefit of the tag embedding is best leveraged by the sentence-bert + agglomerative clustering which is a clear winner for all the three datasets. For the COVID-19 dataset, though the improvement obtained by adding the tag embedding is negligible since the search results for COVID-19 related events are very generic in nature.

Full system evaluation: So far, the assessment for the two components was carried out separately, i.e., the evaluation for the important sentence extraction was based on Level-I annotated data while the evaluation for event coreference resolution was on the basis of Level-II annotations independently. We also conduct the full system evaluation for CWMG and CWAL datasets, i.e., the complete evaluation was only dependent on Level-II annotated data. For this case we trained the GAN-BERT classifier with 30% of the labeled data along with the unlabeled data (discussed in section 4.2), and had predictions for the rest of 70% data. Now, we consider only the *true positives* (labeled as important, and also predicted important), before performing the coreference resolution. This task is evaluated based on the Level-II annotated data. Table 5 shows the comparison between the full system evaluation result and the standard result. The results shown here are the average value of the four different standard metrics (MUC, B^3, CEAF_E and BLANC) corresponding to the best performing unsupervised model. The supervised results are very similar and therefore not shown.

Dataset	Method	Recall	Precision	F1
CWMG	MA	0.65	0.71	0.68
	MP	0.62	0.65	0.63
CWAL	MA	0.60	0.66	0.63
	MP	0.55	0.59	0.57

Table 5: Comparison of full system evaluation result for the standard coreference resolution result. MA: Important sentences obtained through manual annotation, MP: Important sentences obtained from model prediction.

System	CWMG Dataset		CWAL Dataset	
	AR1-F	AR2-F	AR1-F	AR2-F
MM	0.023	0.001	0.052	0.024
DT	0.008	0.001	0.022	0.002
ED + DT	0.015*	0.006*	0.026*	0.002
CLUST	0.028	0.02	0.055	0.040
ED + CLUST	0.034•	0.025•	0.086•	0.071•
Our method	**0.062**†*•	**0.043**†*•	**0.069**†*•	**0.042**†*•

Table 6: Comparison of our method for the with the existing state-of-the-art TLS methods - (1) MM (submodularity based method): Martschat and Markert (2018) and (2) DT: datewise and (3) CLUST: clustering based TLS by Gholipour Ghalandari and Ifrim (2020), ED: Event detection. †, *, • show that our results are significantly different from MM, ED + DT, ED + CLUST respectively. In turn, any method with ED (*, •) is significantly better than MM.

Comparison with TLS: Since our method has some parallels with TLS, in this section we perform a thorough comparison with state-of-the-art TLS systems for the CWMG and CWAL datasets. Note that the output of our system is not similar to that of the standard TLS output. In order to make the comparison possible we added a simple summarization step at the end of our pipeline. We used the BERT extractive summarizer (Miller, 2019) to extracted the two most important sentences as the summary for each of the event clusters generated by our method. We evaluated the summaries using the alignment-based ROUGE (AR) F-Score (Martschat and Markert, 2017). Unlike (Gholipour Ghalandari and Ifrim, 2020), we did not use of any date ranking method to rank the dates of the predicted timeline and compare the ground truth with top-k predicted timeline. We tested all the approaches using our *Level I* annotated data as the ground truth reference. Table 6 shows the detailed comparison of our approach with one of the existing state of the art TLS approaches on several TLS datasets. Our two-step approach clearly outperforms the standard TLS methods for our historical datasets. One possible reason behind the success of our model could be that the sentence selection process for the summary in the standard TLS approaches are highly sensitive to the keywords used for the particular dataset and generating quality keywords for a dataset consisting of diverse events like ours requires domain-expertise (see Table 7). A very crucial observation is that event detection prior to summarization always helps – our method as well as one of the baseline methods (Gholipour Ghalandari and Ifrim, 2020) where event detection can be easily incorporated show significantly[13] improved performance.

[13]Statistical significance were performed using Mann–Whitney U test (Mann and Whitney, 1947)

[1930-04-06] I feel you are right in confining your attention to the salt tax for the time being .	[1930-05-04] In Karachi , Peshawar and Madras the firing would appear to have been unprovoked and unnecessary . Bones have been broken , private parts have been squeezed for the pur- pose of making volunteers give up , to the Government valueless , to the volunteers precious salt .
[1930-04-30] The addressee had been arrested on April 30 , 1930 , during the Vedaranyam Salt Satyagraha . In reply to the addressee 's letter regarding the order of the Madras Government permitting the collector of Tanjore to prosecute the satyagrahis breaking the salt law in the South 2	[1930-04-11] After returning from the Assembly work at Delhi I immediately held confe- rence of Maharashtra National Party and have decided to start and organ-ise
[1930-04-14] I got the book about salt which you sent with Keshavram	[1930-04-14] It is 10.30p.m. Jawahar has also been arrested .Pandya , Ghia and others have been arrested here .If things continue to move with the present velocity , he wo n't have even six months ' rest .I never expected this phenomenal res- ponse.

Table 7: Sample summary generated using (Gholipour Ghalandari and Ifrim, 2020) (left) and our method (right) on the CWMG dataset. Text in blue indicates the portion present in the ground truth timeline.

7 Timeline visualization

Generating a timeline would not be that impactful unless it is visualized in an interpretable and convenient way. The primary features of a timeline, i.e., the flow of events, the temporal and spatial elements and their relationship need to be clearly highlighted in a timeline visualization. We incorporate an elegant visualization for the generated event timelines using *vis-timeline* javascript library (Appendix A.3 shows an example timeline).

Survey: In order to understand the effectiveness of the interface we ran an online crowd-sourced survey. Out of 33 paritcipants with different educational backgrounds, 93% agreed that the interface was very useful for summarization of historical timeline of events. Another intriguing point is that 88% participants found some information which would have been hard for them to fathom just by reading the CWMG plaintext (see more results in Appendix A.4).

8 Conclusion

In this work we presented a framework to generate event timeline from any timestamped document. The entire pipeline has two parts – important event detection and event coreference resolution. We achieve state-of-the-art performance for both these tasks. Our two step method also outperforms several recent TLS baselines. Finally we render the events obtained using a user-friendly visualization tool. Our system is not limited to any actor specific event (human or location) which made the coreference resolution task even more challenging. We believe that our work will open up new and exciting opportunities in history research and education.

9 Ethical considerations

We have framed our datasets by collecting textual information from publicly available online resources and these do not contain any individual private information. The two historical datasets, i.e., the CWMG and the CWAL have been constructed by using the two specific online sources mentioned in 4.1, while the privacy rights have been acknowledged. The contents in the COVID-19 event dataset are collected from freely accessible Wikipedia and publicly available information from `https://who.int`. Finally, the datasets have been annotated by the research scholars and university undergraduate students voluntarily.

References

Sayantan Adak, Atharva Vyas, Animesh Mukherjee, Heer Ambavi, Pritam Kadasi, Mayank Singh, and Shivam Patel. 2020. Gandhipedia: A one-stop ai-enabled portal for browsing gandhian literature, life-events and his social network. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL '20, page 539–540, New York, NY, USA. Association for Computing Machinery.

Alessio Aprosio and Sara Tonelli. 2015. Recognizing biographical sections in wikipedia. pages 811–816.

Amit Bagga and Breck Baldwin. 2000. Entity-based cross-document coreferencing using the vector space model. *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, 1.

David Bamman and Noah A. Smith. 2014. Unsupervised discovery of biographical structure from text. *Transactions of the Association for Computational Linguistics*, 2:363–376.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution.

Harsimran Bedi, Sangameshwar Patil, Swapnil Hingmire, and Girish Palshikar. 2017. Event timeline generation from history textbooks. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pages 69–77, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Leo Born, Maximilian Bacher, and Katja Markert. 2020. Dataset Reproducibility and IR Methods in Timeline Summarization. In *LREC 2020*.

Souvic Chakraborty, Pawan Goyal, and Animesh Mukherjee. 2020. Aspect-based sentiment analysis of scientific reviews. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*.

Zheng Chen, Heng Ji, and Robert Haralick. 2009. A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*, pages 17–22, Borovets, Bulgaria. Association for Computational Linguistics.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2124–2133, Copenhagen, Denmark. Association for Computational Linguistics.

Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 2020. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119, Online. Association for Computational Linguistics.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Abbas Ghaddar and Phillippe Langlais. 2016. Wikicoref: An english coreference-annotated corpus of wikipedia articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

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.

Marti A. Hearst. 1998. Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. Resolving event coreference with supervised representation learning and clustering-oriented regularization.

Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2004. Multinomial naive bayes for text categorization revisited. In *Proceedings of the 17th Australian Joint Conference on Advances in Artificial Intelligence*, AI'04, page 488–499, Berlin, Heidelberg. Springer-Verlag.

H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Moreno La Quatra, Luca Cagliero, Elena Baralis, Alberto Messina, and Maurizio Montagnuolo. 2021. *Summarize Dates First: A Paradigm Shift in Timeline Summarization*, page 418–427. Association for Computing Machinery, New York, NY, USA.

Yaojie Lu, Hongyu Lin, Jialong Tang, Xianpei Han, and Le Sun. 2020. End-to-end neural event coreference resolution.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics.

H. B. Mann and D. R. Whitney. 1947. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50 – 60.

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.

Derek Miller. 2019. Leveraging bert for extractive text summarization on lectures.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification.

Nafise Sadat Moosavi and Michael Strube. 2016. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Berlin, Germany. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Sabarmati Ashram Preservation and Memorial Trust. 2013. The Collected Works of Mahatma Gandhi. `https://www.gandhiheritageportal.org/the-collected-works-of-mahatma-gandhi`. [Online; accessed 22-February-2020].

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Rob Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. pages 28–34.

W. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850.

M. Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17:485 – 510.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. pages 45–52.

W. Zhang, Q. Chen, and Y. Chen. 2020. Deep learning based robust text classification method via virtual adversarial training. *IEEE Access*, 8:61174–61182.

Ye Zhang and Byron Wallace. 2016. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification.

A Appendices

A.1 Sample annotations

Table 8 shows the examples of Level I annotated data (sentence classification) and Table 9 illustrates Level II annotated data (coreference resolution) for some portions in the CWMG dataset.

A.2 Architecture diagram of supervised mention-pair model

Figure 3 represents the model architecture, which is inspired from Barhom et al. (2019).

doc_id	publication date	time	sentence	importance	type
volume43_book_393	1930-05-04T00:00:00+00:00	1930-05-04T00:00:00+00:00	The public have been told that Dharasana is private property .	1	fact
volume43_book_393	1930-05-04T00:00:00+00:00	1930-05-04T00:00:00+00:00	This is mere camouflage .	1	fact
volume43_book_393	1930-05-04T00:00:00+00:00	1930-05-04T00:00:00+00:00	It is as effectively under Government control as the Viceroy 's House .	1	fact
volume43_book_393	1930-05-04T00:00:00+00:00	1930-05-04T00:00:00+00:00	Not a pinch of salt can be removed without the previous sanction of the authorities .	1	fact
volume43_book_393	1930-05-04T00:00:00+00:00	1930-05-04T00:00:00+00:00	It is possible for you to prevent this raid , as it has been play- fully and mischievously called , in three ways :	0	None
volume43_book_393	1930-05-04T00:00:00+00:00	1930-05-04T00:00:00+00:00	by removing the salt tax ; 1 The letter was drafted on the eve of Gandhiji 's arrest .	0	None
volume43_book_393	1930-05-04T00:00:00+00:00	1930-05-05T00:00:00	He was arrested at 12.45 a.m. on May 5 .	1	event
volume43_book_393	1930-05-04T00:00:00+00:00	1930-05-04T00:00:00+00:00	THE COLLECTED WORKS OF MAHATMA GANDHI 2 . by arresting me and my party unless the country can ,	0	None

Table 8: Sample Level I annotation of CWMG dataset.

sentence	importance	type	cluster
The public have been told that Dharasana is private property .	1	fact	1
This is mere camouflage .	1	fact	1
It is as effectively under Government control as the Viceroy 's House .	1	fact	1
Not a pinch of salt can be removed without the previous sanction of the authorities .	1	fact	1
It is possible for you to prevent this raid , as it has been play- fully and mischievously called , in three ways :	0	None	None
by removing the salt tax ; 1 The letter was drafted on the eve of Gandhiji 's arrest .	0	None	None
He was arrested at 12.45 a.m. on May 5 .	1	event	2

Table 9: Sample Level II annotation of CWMG dataset. We only marked the cluster value for the sentences which are marked as important by at least 2 annotators during the level I annotation.

A.3 Sample timeline

After resolving the event coreference, the generated data is used to create the timeline. In order to generate the title for a specific event, we have used BERT extractive summarizer (Miller, 2019). The idea of visualisation was to make the tool accessible to historians as well as run a survey of the utility of the tool in the first place. Figure 4 shows a sample event timeline generated by the tool from the CWMG dataset.

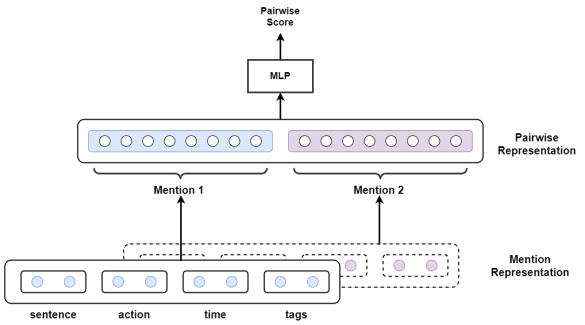

Figure 3: An illustration of the pairwise classification model.

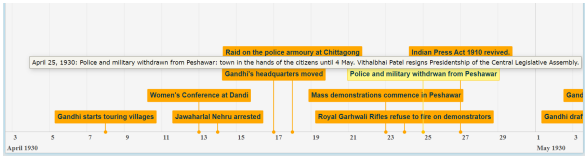

Figure 4: Sample visualization of timeline generated from the CWMG dataset.

A.4 Online survey

In the survey we asked participants a number of questions regarding the readability, correctness and relevance about the information in the generated timeline. 33 participants with various educational backgrounds took part in the survey. 79% of the participants noted that the interface was easily readable. 73% of the total participants reported that they were very satisfied with the overall quality of the automatically generated event timeline summaries.