Machine Reading of Historical Events

Or Honovich[1,*] **Lucas Torroba Hennigen**[2,*] **Omri Abend**[1] **Shay B. Cohen**[3]
[1]The Hebrew University of Jerusalem [2]University of Cambridge [3]University of Edinburgh
or.honovich@gmail.com lft26@cam.ac.uk
oabend@cs.huji.ac.il scohen@inf.ed.ac.uk

Abstract

Machine reading is an ambitious goal in NLP that subsumes a wide range of text understanding capabilities. Within this broad framework, we address the task of machine reading the time of historical events, compile datasets for the task, and develop a model for tackling it. Given a brief textual description of an event, we show that good performance can be achieved by extracting relevant sentences from Wikipedia, and applying a combination of task-specific and general-purpose feature embeddings for the classification. Furthermore, we establish a link between the historical event ordering task and the event focus time task from the information retrieval literature, showing they also provide a challenging test case for machine reading algorithms.[1]

1 Introduction

Machine reading concerns the extraction of entities and relations from text and the ability to use them meaningfully, for instance by answering questions based on them, inferring other relations from them, or using them to compile knowledge bases. Such an inclusive task definition necessarily builds on a wide range of NLP capabilities, from syntactic and semantic analysis, to the use of world knowledge and common sense. The inclusive nature of the task supports the development of general-purpose methods, but also results in low performance in absolute terms, difficulty in defining widely agreed-upon evaluation protocols, and difficulties identifying the sources of prediction errors (Stanovsky and Dagan, 2016; Rajpurkar et al., 2016; Clark et al., 2018).

This paper addresses a sub-task of machine reading, namely the task of estimating when a historical event took place. This distinguishes it from traditional question answering (Rajpurkar et al., 2016) as the answer may not be given in the text but the models should still be able to place events in the correct period of time. In turn, this means that models trained for historical event ordering may have real-world applications such as to serve as a fallback for temporal question answering when the answers are not present in the text and to improve search engines that leverage the implicit time of queries (Gupta and Berberich, 2016).

	Year	Event text
OTD	2005	107 die in Amagasaki rail crash in Japan.
	1939	BMI (Broadcast Music Incorporated) formed.
	1864	General Sherman's armies reach Savannah & 12 day siege begins.
WOTD	1887	Buffalo Bill Cody's Wild West Show opens in London.
	1399	Henry IV is proclaimed King of England.
	1943	First Flight of the Gloster Meteor, Britain's first combat jet aircraft.

Table 1: Entries from the OTD and WOTD datasets.

Concretely, given a short text description of a historical event, and an external data source (henceforth *contextual information* or CI), the task is to predict the year in which the event happened. The external source in our case is Wikipedia. For example, given the event description "The Government of Turkey expels Patriarch Constantine VI from Istanbul," the task is to infer the year it took place (i.e., 1925). We select Wikipedia as a source for contextual information, due to its broad coverage, and the wide interest it receives in the NLP community. Indeed, Wikipedia has often featured as a semi-structured knowledge base, e.g., as a source of concept grounding (Bunescu and Paşca, 2006) and indirect supervision (Mintz et al., 2009).

We hypothesize that aside from time expressions, the CI words themselves give an approximate time in which an event happened. For example, the pres-

[1]Code and data are available at https://github.com/ltorroba/machine-reading-historical-events.

[*]Equal contribution.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7486–7497

ence of the word "spacecraft" in the CI probably indicates an event that occurred after 1900, while the presence of the word "sword" most likely indicates an event that occurred before 1900. The task is therefore different from tasks addressing the extraction and normalization of time expressions, or from related tasks pursued in the context of information retrieval (see §8). Our results support this hypothesis, and demonstrate that even when time expressions are not present in the data, it is still possible to predict the approximate year in which an event happened.

We compile two datasets for the task, based on the websites "Wikipedia On This Day" webpages (WOTD) and "On This Day" (OTD). We consider WOTD as an in-domain setting, given that it is taken from Wikipedia as well (albeit from an entirely disjoint part of Wikipedia). The OTD setting was selected to be maximally challenging for leveraging external data sources, since (1) event descriptions are taken from a different website, and may be formulated very differently from Wikipedia; (2) it is an order of magnitude larger, and so the classifier has plenty of data to train on, even without relying on external data sources.

Our results show that on WOTD, good performance can be obtained by detecting relevant sentences from Wikipedia and extracting year mentions in them, but that substantially better performance can be reached when additionally encoding the entire sentences, using neural machinery. In OTD, CI yields more modest improvements. Results in absolute terms are high: the best models obtain a mean Kendall's τ correlation with the correct event ordering of 0.77 (WOTD) and 0.71 (OTD).

2 Task Definition and Motivation

The historical event ordering (HEO) task is defined as follows. Given a set of brief event descriptions and some textual resource, the task is either to predict the year in which each event occurred, or to find a ranking of events such that they are ordered by date of occurrence. The first variant is stronger than the second, as it implies a ranking. Our evaluation uses both rank correlation (Kendall's τ) and measures of the distance between the year the event took place and the predicted year. See Section 5.2 for details.

Differences from Question Answering. While traditional question answering tasks require the answer to be in the text (e.g., Hermann et al., 2015; Rajpurkar et al., 2016), the HEO task is based on estimating the time of occurrence of an event. This estimation is based solely on lexical cues, and does not require an explicit answer in any text. This is a major advantage of HEO models, as explicit answers are not always present in the text for two reasons: (i) we would need a massive amount of text for good coverage of historical events, which may be unfeasible to use in real-world applications; and (ii) new events are constantly occurring, and existing machine reading comprehension models will invariably fail on those (e.g., "When was Donald Trump elected president?" will not be covered in old data, but could be inferred to have happened recently based on recognizing the named entity "Donald Trump"). As answers are not guaranteed to be in the text, the HEO task is somewhat more challenging than traditional question answering tasks. The task's challenge is also evidenced in that it requires temporal commonsense reasoning and in being challenging for humans (see §6).

Real-World Applications. As previously mentioned, HEO models do not assume the presence of the answer in the source text, and can thus be used for temporal question answering when the answers are not present in it. By leveraging the lexical information that exists only in the question itself, these models can serve as a fallback for such cases. Other possible applications are dating of historical documents based solely on the documents' text, improving search engines that leverage the time of queries (Gupta and Berberich, 2016), as well as making inferences that involve rough temporal placement of the statement (e.g., inferences involving refrigerators are unlikely to be relevant before the 20th century).

3 Data Collection

This work introduces two datasets: WOTD and OTD. Despite the similarity in their names, we are not aware of any influence or other relation between them. Using both datasets thus makes our experimental analysis less prone to be biased by dataset-specific artifacts.

Wikipedia On This Day (WOTD) was scraped from Wikipedia's On this day webpages.[2] The dataset contains 6,809 entries. Some example entries are presented in Table 1. Events in Wikipedia's On This Day pages are crowdsourced,

[2] E.g., https://en.wikipedia.org/wiki/Wikipedia:On_this_day/Today, accessed 03/2018.

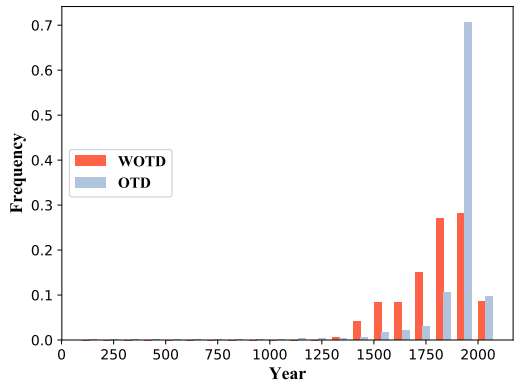

Figure 1: Distribution of event years in the OTD and WOTD, binned into bins of 100 years. The y-axis corresponds to the proportion of events falling into the bin.

but must adhere to specific guidelines[3] which include the validity and overall relevance of the historical event. The earliest label in this dataset is 1302, and the latest is 2018. The median year is 1855.0 whereas the mean is 1818.7. The standard deviation is 156.5 years.

On This Day (OTD) is a scrape of the On This Day – Today in History, Film, Music and Sport (Li, 2018).[4] On This Day has a dedicated team that adds, verifies content, and responds to corrections from the public.[5] The dataset contains 75,135 entries consisting of a sentence describing the event and the event's date. We removed 96 events from the original dataset, which happened BCE (Before Common Era), and also removed events that had not happened yet. The earliest event in the dataset occurred on year 1 CE (Common Era), and the latest occurred in 2018 CE. The median label is 1960.0, while the mean label is 1913.8, so the distribution of labels is not uniform: there are more events occurring in recent times. The standard deviation for the labels is 172.3 years.

Examples of entries from OTD are presented in Table 1. We note that the overwhelming majority of events in the datasets are real historical events, and though we did not conduct an exhaustive analysis, the only two we identified as fictional were removed by our filters. There are 8 events that are dated in the future, and all but one of those ("Earth's 1st contact with the extra-terrestrial Vulcan species in the Star Trek universe", on 2063 CE) correspond to either calendar occurrences (e.g., "Beginning of 2nd Julian Period (1/1 OS)", on 3268 CE) or astronomical events (e.g., "Comet Swift-Tuttle approaches close to Earth", on 2126 CE). Our pruning strategy (discard events before 1 CE) was deliberately aggressive, removing 88 events including widely accepted ones (e.g., "Battle of Actium", on 31 BCE); however, it is also effective in removing potentially fictitious events (e.g., "Creation of the world begins according to the calculations of Archbishop James Ussher", on 4004 BCE) or whose exact date may not be known (e.g., "Battle of Megiddo" dated to 1457 BCE, but subject to debate).

Figure 1 shows the distribution of event years in OTD and WOTD. Both datasets have significantly more recent events from the last few centuries. We use a random 80/10/10 split of each dataset to form the training, validation and test sets.

4 Algorithmic Approaches

We propose two models: a bag of embeddings model (BOE) and a recurrent neural network model (LSTM). Both take a training example and output a timestamp, in our case the year of the event. We explore two supervised settings: a classification setting, where each possible year corresponds to a different class, and a regression setting, where the labels are the numerical value of the timestamp.

As baselines, we define two models: one predicts the mean year of the training set (MEAN), and one predicts the median year present in the extracted CI, falling back to the other baseline if no years are found (CIYEAR).

4.1 Retrieving Relevant Wikipedia Sentences

Key Entities And Actions. We first identify the key entities and actions in each event description. Concretely, for a given event description e, we define its *key entities* to be phrases from e that are likely to be the topic of a Wikipedia article that contains information relevant to e. We define *key actions* to be a tuple of all verbs in e, excluding some aspectual (e.g., "begin") and auxiliary verbs. We lemmatize all key actions.

For example, given the event description "The Sixth Coalition attacks Napoleon Bonaparte in the Battle of Leipzig", we mark ("Sixth Coalition",

[3] https://en.wikipedia.org/wiki/Wikipedia:Selected_anniversaries, accessed 04/2020.

[4] https://www.onthisday.com, accessed 01/2019.

[5] https://www.onthisday.com/about.php, accessed 04/2020.

"Napoleon Bonaparte", "Battle of Leipzig")[6] as the key entities, and "attack" as the key action.

Entities and actions are extracted using a set of pre-defined rules, based on linguistic features such as part-of-speech (POS) tag, syntactic dependency labels, and entity type, for words recognized as named entities. Linguistic features, including named entities, are extracted using spaCy.[7] Some example rules for detecting key entities are:

1. Take all named entities, excluding some entity types such as MONEY, PERCENT and ORDINAL.
2. Take all nominal subjects, except pronouns and nominalized adjectives. For example, for "The Sixth Coalition attacks Napoleon Bonaparte in the Battle of Leipzig", "Sixth Coalition" is marked as a key entity.

The complete set of rules can be found in the supplementary material. The majority of key entities are named entities and are therefore identified by the first rule above.

Article Retrieval. We use the extracted key entities to retrieve relevant Wikipedia articles. For each key entity, we retrieve the first search result returned for the entity name, as proposed by the Wikipedia API. We use the Python Wikipedia library[8] for performing the queries.

Sentence Filtering. Filtering seeks to identify sentences related to the historical event in question. For example, for the event "The Skye Bridge is opened", the sentence "Construction began in 1992 and the bridge was opened by Secretary of State for Scotland Michael Forsyth on 16 October 1995" from the article "Skye Bridge" is relevant.

We denote by $\{t_1, ..., t_k\}$ the key entities for each event, where k varies from one event to another, and test several filtering methods:

1. Sentences from an article with title t_i that contain one or more t_j for j $\neq$ i, and a key action.
2. Sentences from an article with title t_i that contain one or more t_j for j $\neq$ i.
3. Sentences from an article with title t_i that contain all t_j for j $\neq$ i.
4. Sentences that contain a date.
5. Sentences from an article with title t_i contain one or more t_j for j $\neq$ i, and a date.

Article sections with headers such as "See also" and "Bibliography" are removed.

Following a manual inspection of the extracted sentences with each of the methods, we find the following method works best: (1) find all sentences according to the first filter; (2) if no relevant sentences are found, apply the second filter instead. In addition, we add the original textual description of the event (taken from OTD/WOTD) to the list of relevant sentences.

Extracting Year Mentions. Given the relevant sentences for each event, we extract from them all year mentions. Years are extracted using the following method: first, we use named entity recognition to extract all dates. Second, of the words recognized as dates, we keep only those whose POS tag is NUMBER.[9] We then parse the dates and extract years, using a simple rule-based parser.[10] We present here some statistics regarding years extracted for the WOTD validation set. For 1.8% of the events, the real year appeared in the event title itself. For 59.5% of the events, at least one year appeared in the contextual information extracted from Wikipedia. Out of the events for which at least one year was extracted, 59.5% had the correct year in the extracted information. In total, for 35.4% of the events, the correct year appeared in the contextual information extracted.

4.2 Baseline Models

To obtain an estimate of the difficulty of the task, we design two baseline models. The MEAN model predicts the mean year seen in the training set, adding Gaussian noise $\epsilon \sim \mathcal{N}(0, 1yr)$ to break ties and induce an ordering. The CIYEAR model extracts year mentions, as detailed above, and predicts the median of all extracted years entities. If no years are found, the model defaults to MEAN.

4.3 Bag of Embeddings Model

We use two types of features: (1) the average of the word embeddings for all lemmatized words in the extracted sentences, excluding stop words and punctuation (as defined by SpaCy); (2) the median value of all year mentions. To represent the median year, we use one-hot encoding for the tens,

[6] In some cases, overlapping entities are extracted. During the next step of extracting Wikipedia's articles, we remove duplicate articles.

[7] www.spacy.io. We used spaCy's v2 "en_core_web_lg" model.

[8] www.pypi.org/project/wikipedia

[9] Again, all linguistic features are extracted using spaCy.

[10] www.pypi.org/project/python-dateutil

hundreds and thousands of the median year and concatenate this encoding to the average embedding. We experimented with encoding the least significant digit as well, but find this lowers results. We explore two variants of the model:

Classification. In the classification setting, the final module consists of a multilayer perceptron (MLP), where class labels are the target years. We note that in the classification settings, the predicted years can only be those that appear in the training set. Since most of our evaluation metrics do not require an exact prediction, but rather an approximate prediction, the classification still yields good results. The final layer is a softmax layer, and the loss function used is log-loss.

Regression. In the regression setting, the network architecture is an MLP with a single output. The regression target is the year of occurrence. The loss function used is L1 loss. We experimented with mean squared error loss (L2) as well, but this gave lower performance.

4.4 Long Short Term Memory Model

The LSTM model takes as input the tokens for the event text and the extracted sentences. A bidirectional LSTM (Hochreiter and Schmidhuber, 1997; Graves et al., 2005) is used to compute an encoding of the event sentence ($\mathbf{e}$) and each CI sentence ($\mathbf{c}_1, \ldots, \mathbf{c}_n$). We then use an attention mechanism (Bahdanau et al., 2015) to compute a similarity score between the event sentence and each CI sentence, and compute an attention-weighted average of the CI encodings, $\mathbf{c}'$. When training models with CI, we concatenate both $\mathbf{e}$ and $\mathbf{c}'$ and use that as input to an MLP that performs the final year prediction. When not using CI, the only input to the MLP is $\mathbf{e}$. The structure of the MLP depends on whether the model is operating on a classification or a regression setting. The two variants we explore are:

Classification. In the classification setting, the final module is composed of an MLP that computes the logits of the event happening in a specific year. All years between the minimum and maximum year present in the training set are valid targets. We minimize the cross-entropy loss of the predicted year.

Regression. In the regression setting, the final module consists of an MLP with a single output. The regression target is the normalized year of the event. We normalize by subtracting the mean year of the training set and dividing the result by the standard deviation. We experimented with regression to unnormalized targets, but found this degraded performance. We minimize the L2 loss of the predicted year.

	Setting		LSTM		MLP	
	CI Use	Mode	L	D	L	D
WOTD	No CI	Regression	3	200	3	50
		Classification	3	200	3	300
	CI	Regression	2	200	3	200
		Classification	2	100	1	50
OTD	No CI	Regression	3	300	2	100
	CI		2	200	2	100

Table 2: Setting-specific hyperparameter values for the LSTM model. L = Layers, D = Layer dimensionality.

5 Experimental Setup

In this section we describe our experimental setup and the evaluation metrics we use.

5.1 Hyperparameters and training

For the BOE model, in the classification setting we set it to have two hidden layers, each with 1000 neurons. We ran experiments with Glove (Pennington et al., 2014) and FastText (Bojanowski et al., 2016) word embeddings and found that Glove vectors with dimension 300, pretrained on Wikipedia 2014, performed best. The initial learning rate of the MLP is set to 0.001. We use L2 regularization with $\alpha = 10^{-4}$. In the regression setting the model has one hidden layer with 32 units. We use Glove with dimension 300. The initial learning rate is set to 0.01. In both settings, we use ReLU as an activation function and Adam for an optimizer (Kingma and Ba, 2014). We experimented with L1 and L2 regularization but found that this doesn't improve performance.

We found the LSTM model to be sensitive to hyperparameter values, and therefore tuned it individually for each setting. The final hyperparameters are shown in Table 2. We use the Adam optimizer (Kingma and Ba, 2014) with $\eta = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and use PReLU activations (He et al., 2015) in the MLP. We train for a maximum of 100 epochs, doing early stopping if the validation loss has not improved in 25 epochs. Furthermore, we decay the learning rate by a factor of 0.1 if there is no reduction in validation

set loss for 10 epochs. Preliminary experiments with Glove (Pennington et al., 2014), ELMo (Peters et al., 2018) and FastText (Bojanowski et al., 2016) word embeddings showed that concatenating 200-dimensional Glove and 300-dimensional FastText embeddings performed best. We experimented with L2 regularization and dropout on both the MLP and LSTM but found that the performance improvement was negligible, and so we did not use them for our final experiments. Our LSTM implementation was done using AllenNLP (Gardner et al., 2018). All hyperparameter tuning was done against the development data.

5.2 Evaluation Metrics

Kendall's Tau (τ), formally Kendall's rank correlation coefficient (Kendall, 1938, 1945), is a standard metric used to measure two different rankings of the same set. Formally, for two rankings X and Y, the form of a general correlation coefficient (Daniels, 1944) is

$$\tau = \frac{\sum_{i,j=1}^{n} a_{ij} b_{ij}}{(\sum_{i,j=1}^{n} a_{ij}^2)(\sum_{i,j=1}^{n} b_{ij}^2)}, \qquad (1)$$

where a_{ij} is the score given to a pair (X_i, X_j) and b_{ij} to the pair (Y_i, Y_j). For Kendall's τ, $a_{ij} = 1$ if $X_i < X_j$ and $a_{ij} = -1$ if $X_i > X_j$, and similarly for b_{ij} and Y. In plain words, τ is the number of pairs which X and Y order in the same way minus the number of pairs that are not ordered in the same way, divided by the total number of pairs. For the case where there are no ties, Kendall's τ is a shifted and scaled version of pairwise accuracy, where $\tau = -1.0$ corresponds to zero accuracy and 1.0 to perfect accuracy. To accommodate for ties, we set $a_{ij} = 0$ when $X_i = X_j$, and $b_{ij} = 0$ when $Y_i = Y_j$, as described by Kendall (1945). This has the same effect as replacing tied members in each set with all permutations of a contiguous set of integer ranks and averaging by the total number of permutations.

Exact Match. Percentage of events in which the predicted year exactly matches the gold-standard.

Mean Absolute Error. This is the absolute mean error for the predictions, in years.

Distance under 20Y and 50Y. The percentage of events whose prediction error was under 20/50 years.

6 Results

Table 3 presents the results of our experiments. We report the average of each statistic over 6 runs, alongside the standard error of the mean at 95% confidence. We include a detailed comparison of the different architectures on the WOTD dataset. We additionally select the best performing BOE and LSTM models on the WOTD development set and train them on the OTD dataset.

Our results show that the Wikipedia enrichment is an essential component of the protocol. For the WOTD dataset, all models exhibit a statistically significant improvement in ordering when adding CI, with the smallest improver being the LSTM classification model, with a $+0.053$ change in τ, and the largest improver being the BOE regression model, with a change $+0.098$ in τ.

For the OTD dataset, the LSTM model showed a modest but statistically significant improvement when adding CI. The BOE model presents a minor decrease in performance; however, we obtain a statistically significant improvement of $+0.027$ in τ by restricting the CI to only include year mentions.[11] As the OTD and the extracted CI are from different domains, the words of the contextual information most probably add too much noise for the BOE model to handle, which is why a performance improvement is observed when only including years, which are not domain-specific. This indicates that leveraging CI is important, even in this more challenging scenario, where the training data is large and the CI is from another domain, but also suggests that additional improvements, such as using domain adaptation techniques (Ziser and Reichart, 2017) for bridging the domain difference, are required to obtain better performance.

One difference between the regression and classification settings is that the latter has higher exact match metrics than the former. This reflects the nature of the two architectures: when using L1/L2 regression, the loss is proportional to the difference in the prediction, whereas in classification what matters is the probability assigned to the exact year.

On the whole, the LSTM model produces better predictions than the BOE model, according to most measures. This is perhaps unsurprising, as it is able to capture word context when analyzing the inputs, leading to more effective reasoning.

[11] This experiment gave the following results: KT - 0.615 $\pm$ 0.002, EM - 10.8 $\pm$ 0.2, 20Y - 67.6 $\pm$ 0.3, 50Y - 84.4 $\pm$ 0.2, MAE - 36.7 $\pm$ 0.4.

Dataset	CI use	Mode	Model	Accuracy				MAE
				KT	EM	20Y	50Y	
WOTD	No CI	Regression	BoE	0.564 ± 0.008	0.2 ± 0.1	16.8 ± 1.1	40.8 ± 1.5	83.6 ± 2.3
			LSTM	**0.688 ± 0.004**	2.2 ± 0.6	42.7 ± 1.4	**68.8 ± 0.3**	**51.6 ± 0.5**
		Classification	BoE	0.627 ± 0.007	**10.9 ± 0.5**	**49.8 ± 0.6**	66.3 ± 0.5	59.3 ± 1.0
			LSTM	0.639 ± 0.021	3.5 ± 0.5	39.4 ± 1.1	61.6 ± 1.6	64.5 ± 2.3
		Baseline	MEAN	0.005	0.3	9.40	28.2	127.2
	CI	Regression	BoE	0.662 ± 0.008	1.9 ± 0.4	51.4 ± 1.0	67.4 ± 0.4	53.4 ± 1.0
			LSTM	**0.767 ± 0.008**	2.1 ± 0.8	51.5 ± 4.2	**77.0 ± 2.0**	**38.4 ± 2.0**
		Classification	BoE	0.705 ± 0.009	9.1 ± 0.6	**61.9 ± 0.9**	74.8 ± 0.5	45.8 ± 0.7
			LSTM	0.692 ± 0.010	4.1 ± 0.4	44.6 ± 1.9	65.9 ± 1.6	56.9 ± 2.2
		Baseline	CIYEAR	0.551	**13.1**	56.2	66.7	64.5
OTD	No CI	Classification	BoE	0.588 ± 0.003	**11.1 ± 0.2**	65.1 ± 0.2	83.1 ± 0.3	38.2 ± 0.4
		Regression	LSTM	**0.683 ± 0.007**	3.0 ± 0.3	**67.8 ± 0.8**	**87.3 ± 0.3**	**29.0 ± 0.4**
		Baseline	MEAN	0.006	0.5	12.9	37.4	85.7
	CI	Classification	BoE	0.560 ± 0.005	**10.0 ± 0.2**	64.4 ± 0.3	82.3 ± 0.2	40.3 ± 0.8
		Regression	LSTM	**0.707 ± 0.005**	3.2 ± 0.2	**70.5 ± 0.6**	**88.9 ± 0.2**	**26.7 ± 0.6**
		Baseline	CIYEAR	0.323	6.3	41.7	62.5	60.1

Table 3: Comparison of the BoE and LSTM models under classification (Classification) and regression (Regression) settings, with and without contextual information (CI) on the WOTD dataset (top), along with results from best BoE and LSTM models on OTD dataset (bottom), with and without contextual information. We include a 95% confidence interval of the mean of each metric computed over 6 runs. Best models on the OTD dataset were picked from the WOTD development set. We also include our baseline scores (Baseline). KT = Kendall's Tau, EM = Exact Match, 20Y = Distance under 20Y, 50Y = Distance under 50Y, MAE = Mean Absolute Error.

Ablation study. Table 4 presents the results of two ablation studies on the best performing models on the WOTD development set, which are the LSTM regressor and BoE classifier. Both studies are conducted on the WOTD development set. To save space, we omit confidence intervals, but a table including those can be found in the appendix.

Study A was conducted only on datapoints from the WOTD dataset with contextual information. We observe that for both models, the impact of removing the event text and using only the extracted contextual information leads to a τ change of -0.043 for BoE and -0.071 for LSTM. This shows that the heuristics we propose for extracting CI are effective at retrieving relevant information.

Study B was conducted on all datapoints from the WOTD dataset. We report the impact of removing tokens denoting years, dates and numbers from both the CI and the event text. We remove years using the method described in §4.1. We remove dates by removing any tokens within a DATE entity. We remove numbers using the *like_num* property of the spaCy tokenizer, which includes different forms that may be considered numerical (e.g. "1" and "one"). Clearly, the removal of dates subsumes the removal of years, and we expect the removal of numbers to remove at least part of all dates, including years, alongside other date-unrelated numbers.

The removal of these features has a very similar impact. This is particularly the case for the LSTM model, where the change in τ was -0.041, -0.042 and -0.051 when removing years, dates and numbers, respectively. BoE presents similar differences in performance when removing those features, with a change in τ of -0.045/-0.031/-0.054 when removing years/dates/numbers. These results support our hypothesis that substantial information about the time of an event is encoded in the vocabulary used, and not only in the time expressions.

Human Performance We compare our results to human performance on this task. Three participants were given 100 randomly selected events from the WOTD dataset and were asked to predict years of occurrences, without using any contextual information. All participants consider themselves as having good knowledge of history, but are not history experts. On average, their error was 52.3 years. The participant who had the best results had a mean error of 34.6 years, which is only 3.8 years less than our best result on the WOTD dataset.

7 Qualitative Analysis

In order to demonstrate the challenges put forth by the addressed task, we examine some events from the OTD development dataset on which our best performing models, LSTM regressor and BoE classifier, got significant prediction errors.

We observe that some events contain words that are usually associated with a different period in time than the year the event occurred in. For exam-

	Model	Accuracy				
		KT	EM	20Y	50Y	MAE
A	BOE	0.674	7.9	49.6	63.1	48.2
	– event text	0.631	7.0	48.2	60.6	54.6
	LSTM	0.765	1.8	50.2	78.8	39.0
	– event text	0.694	1.3	39.9	69.7	50.6
B	BOE	0.668	9.1	55.3	71.0	50.7
	– years	0.623	7.4	46.1	65.0	61.1
	– dates	0.637	7.4	46.7	64.2	60.5
	– numbers	0.614	8.0	46.3	64.2	63.1
	LSTM	0.774	1.8	50.4	77.3	39.9
	– years	0.733	1.3	41.0	68.5	48.3
	– dates	0.732	1.4	42.5	70.1	47.7
	– numbers	0.723	1.3	40.1	68.8	49.1

Table 4: Ablation study for BOE and LSTM models. Study A was conducted only on datapoints from the WOTD dataset with contextual information, and we report the impact of removing the event text from both models. Study B was conducted on all datapoints from the WOTD dataset, and we report the impact of removing tokens denoting years, dates and numbers. Tokens of these types were removed from the event text and from the contextual information. KT = Kendall's Tau, EM = Exact Match, 20Y = Distance under 20Y, 50Y = Distance under 50Y, MAE = Mean Absolute Error.

ple, "Portuguese expel Jesuits" occurred in 1911, but most Jesuits-related events in our training data occurred in the 16th century. One of these events which is particularly similar to the above is "English parliament expels Jesuits", which is dated to 1584. Probably for these reasons the LSTM and BOE had similar outputs for this event – 1559 and 1581, respectively. Another example for such an event is "Order of Merit instituted by King Edward VII", which occurred in 1902, but the word "King" normally appears in events dated to earlier centuries. The LSTM model output for this event is 1527, and BOE model output is 1639. Both events had no CI extracted for them, therefore the models had to rely on words in the event description only.

An example for which relevant CI was extracted but the models still erred substantially is the event "All female jury hears case of Judith Catchpole accused of killing her child (acquit her) in Patuxent County, Maryland". This event is dated to 1656, but the BOE model prediction for the event is 1957, and the LSTM model prediction, 1873, is only slightly better. The contextual information extracted for this event was "Upon her arrival she was accused of several crimes, resulting in a trial on September 22, 1656 in the General Provincial Court in Patuxent County, Maryland". The exact date of occurrence does appear in the extracted data, and still both models have a substantial prediction error. This is probably due to the fact that our training data contains many "court" and "jury" related events, where most events containing "court" are relatively recent (19th century and later), and almost all "jury" related events are dated after 1900.

In some cases, the extracted CI can mislead our models. For the event "Scotland and France form an alliance, the beginnings of the Auld Alliance, against England" that occurred in 1295, LSTM predicted the year 1659. Five sentences were extracted for this event, which contained the years 1603 and 1707. Another example is "Over 250 years after their deaths, William Penn and his wife Hannah Callowhill Penn are made Honorary Citizens of the United States" occurred in 1984. The CI extracted includes the exact true date of the event, but also includes information regarding the Penns' lives, and contains years ranging between 1680 to 1726. This is probably the cause of error for the BOE model, which predicts the year 1721, whereas the LSTM model may have been able to better filter the correct CI, and predict the year 1921.

Errors can also arise from terms that are ambiguous between time periods. "Queen Elizabeth" is such a term: it can indicate an event from the 16th century, but also an event from the 20th/21st centuries. Indeed, we notice confusion of the BOE model on events related to Queen Elizabeth. For example, "Francis Drake knighted by Queen Elizabeth I aboard Golden Hind at Deptford" occurred in 1581, but the BOE model predicts the year 2013 – even though the true target year appears in the extracted CI for the event: "I visited the royal dockyard on 4 April 1581 to knight the adventurer Francis Drake." Similarly, the event "Ted Hughes is appointed British Poet Laureate by Queen Elizabeth II" occurred in 1984, but the BOE model predicts the year 1579, which corresponds to Queen Elizabeth I. We note that for those two events the LSTM model gave better predictions (1566 for the first event and 1981 for the second), which may be related to the inherent difficulty of BOE to address multi-word expressions like "Queen Elizabeth I".

8 Related Work

Work on event ordering can largely be categorized into event ordering in context, which aims to order event instances within a given text or discourse and is tackled as part of the TempEval shared tasks (UzZaman et al., 2013), and lexical event order-

ing (Abend et al., 2015), which attempts to order event types by their prototypical temporal order. Somewhat in between these lines of work is cross-document event ordering (Minard et al., 2015), which orders events that are mentioned across different documents. However, this task does not rely on machine reading external textual resources as we do here, and does not focus on historical events that by their nature are described in a variety of (often incompatible) ways.

A related line of work to ours was pursued in the context of information retrieval (IR). Jatowt et al. (2013) tackled the task of estimating what the "focus time" of a given document is. Focus time is defined as the time to which the main event addressed by the document refers to. They do so by computing the association of words and time expressions, based on their co-occurrence, using a bag-of-words method.

Das et al. (2017) address the task of focus time prediction for short event descriptions, which resembles the task at hand. They do so by using cosine similarity to rank a set of candidate years for each event, all of which are computed using word embeddings. In a similar vein, Morbidoni et al. (2018) find the focus time of short event descriptions by relying on year mention statistics in related Wikipedia articles and DBpedia entries.

While these two works are related to our task in spirit, our work is not an instance of the event focus time (EFT) task. In fact, we believe the EFT task can be seen as a special case of the HEO task. This is evidenced by the approach of EFT systems, which exhibits traditional IR design and techniques, such as producing a ranking of candidate predictions for each document, and is evaluated using ranking-specific metrics that forbid system designs such as predicting years using regression. As HEO subsumes EFT, we attempted to evaluate the performance of EFT systems in the HEO task, but have been unable to obtain code for either of the systems. We have also been unable to reimplement the systems: (Das et al., 2017) leaves implementation details unspecified, and (Morbidoni et al., 2018) utilizes a proprietary system.

Another related line of work seeks to create timelines of temporal events by predicting their starting and ending points. McClosky and Manning (2012) address the problem of ensuring semantically consistent timelines by finding patterns in the ordering of endpoints of different event types, which adds a common sense reasoning component to the system. Leeuwenberg and Moens (2018) construct a relative timeline of events directly, which allows them to circumvent typical pitfalls of pair-wise classifiers, such as computationally intractable inference and constructing globally inconsistent orderings (with cycles). Our work takes a similar approach but instead is able to construct an absolute timeline for the restricted domain of historical events.

Within the domain of temporal text understanding, the extraction and normalization of temporal expression may inform the task at hand. For example, Kuzey et al. (2016) defined the task of tagging temporal expressions, which are named events or facts with temporal scope, such as "second term of Angela Merkel". They used a rule-based system to detect such expressions in free-text and map these expressions to a knowledge base (KB) containing time scopes of temporal events and facts. This approach requires the existence of KB records containing time scopes for the events.

9 Conclusion

In this paper we argued that the task of predicting the time of historical events strikes a balance between being a focused task, with transparent evaluation and interpretable results, and presenting challenges that are not simple to overcome using standard NLP models. We outlined a procedure to extract the CI related to an event and compared two approaches for the task, using bag of embeddings and an LSTM, showing that the latter achieves the best performance. Future work will explore the use of domain adaptation techniques to enhance performance where the domains of the CI and event text differ substantially.

Acknowledgments

We thank the anonymous reviewers for helpful feedback. We would also like to thank Maximin Coavoux, Simone Teufel, and Ryan Cotterell for their help and comments. We gratefully acknowledge the support of Bloomberg (Cohen). This work was partially supported by the Israel Science Foundation (grant No. 929/17)

References

Omri Abend, Shay Cohen, and Mark Steedman. 2015. Lexical event ordering with an edge-factored model. In *Proc. of NAACL*, page 1161–1171.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proc. of EACL*, pages 9–16.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Henry E. Daniels. 1944. The relation between measures of correlation in the universe of sample permutations. *Biometrika*, 33(2):129–135.

Supratim Das, Arunav Mishra, Katsumi Tanaka, and Vinay Setty. 2017. Estimating event focus time using neural word embeddings. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management, CIKM*, pages 2039–2042.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*, ICANN'05, pages 799–804, Berlin, Heidelberg. Springer-Verlag.

Dhruv Gupta and Klaus Berberich. 2016. Diversifying search results using time. In *Advances in Information Retrieval*, pages 789–795, Cham. Springer International Publishing.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *arXiv:1506.03340 [cs]*. ArXiv: 1506.03340.

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

Adam Jatowt, Ching-Man Au Yeung, and Katsumi Tanaka. 2013. Estimating document focus time. In *Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management*, pages 2273–2278.

Maurice G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Maurice G. Kendall. 1945. The treatment of ties in ranking problems. *Biometrika*, 33.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Erdal Kuzey, Vinay Setty, Jannik Strötgen, and Gerhard Weikum. 2016. As time goes by: Comprehensive tagging of textual phrases with temporal scopes. In *WWW '16 Proceedings of the 25th International Conference on World Wide Web*, pages 915–925.

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. *CoRR*, abs/1808.09401.

Xue Li. 2018. Temporal ordering of historical events. Master's thesis, University of Edinburgh.

David McClosky and Christopher D. Manning. 2012. Learning constraints for consistent timeline extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Ruben Urizar. 2015. Semeval-2015 task 4: Timeline: Cross-document event ordering. In *9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Christian Morbidoni, Alessandro Cucchiarelli, and Domenico Ursino. 2018. Leveraging linked entities to estimate focus time of short texts. In *Proceedings of the 22Nd International Database Engineering & Applications Symposium*, IDEAS 2018, pages 282–286, New York, NY, USA. ACM.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In **SEM-SemEval '13*, pages 1–9.

Yftah Ziser and Roi Reichart. 2017. Neural structural correspondence learning for domain adaptation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 400–410.

A Key Entities Extraction

We introduce the full set of rules for key entity extraction. The following are considered key entities:

1. Named entities, excluding the following labels: MONEY, TIME, PERCENT, DATA, ORDINAL, QUANTITY, CARDINAL.

2. A compound whose head, or the head of its head, is the root of the sentence. For example, in the event: "Apollo program: Apollo 14 returns to Earth after the third manned Moon landing.", this rule will extract "Apollo program".

3. An adjectival modifier or nominal modifier whose head is the root, where the root is not a verb. For example, in the event "Mexican–American War: The first large-scale amphibious assault in U.S. history is launched in the Siege of Veracruz.", this rule will extract "Mexican–American War".

4. All nominal subjects, except pronouns and nominalized adjectives. An example for this rule can be found in the paper itself.

5. All passive nomial subjects that are proper nouns. For example, in the event "The USS George Washington is launched. It is the first nuclear-powered ballistic missile submarine.", this rule extracts "USS George Washington".

6. All direct objects that are proper nouns.

We note that for all rules except the first we take all the sub-tree to which the word we found belongs. We remove from the sub-tree determiners, puncutation and adverbs. In the example given above – "The USS George Washington is launched. It is the first nuclear-powered ballistic missile submarine." – the word that complies to the fifth rule above is "Washington", but we extract "USS George Washington". In addition, we remove relative clauses of the phrase. For example, in the event "The British Parliament passes the Stamp Act that introduces a tax to be levied directly on its American colonies.", we are interested in extracting "Stamp Act", but we leave out the part "that introduces a tax...". Similarly, from the event "The debut exhibition of the Belitung shipwreck, containing the largest collection of Tang dynasty artifacts found in one location, begins in Singapore.", we leave out the part "containing the largest..." when extracting "debut exhibition of Belitung shipwreck".

B Extended Ablation Results

Refer to Table 5 for an expanded table of the ablation experiments that includes metric error.

	Model	Accuracy				
		KT	EM	20Y	50Y	MAE
A	BOE	0.674 ± 0.010	7.9 ± 0.7	49.6 ± 0.6	63.1 ± 0.4	48.2 ± 1.8
	– event text	0.631 ± 0.009	7.0 ± 0.6	48.2 ± 0.7	60.6 ± 0.7	54.6 ± 1.1
	LSTM	0.765 ± 0.009	1.8 ± 0.4	50.2 ± 4.6	78.8 ± 1.9	39.0 ± 2.1
	– event text	0.694 ± 0.011	1.3 ± 0.4	39.9 ± 5.1	69.7 ± 4.3	50.6 ± 3.3
B	BOE	0.668 ± 0.007	9.1 ± 0.4	55.3 ± 0.8	71.0 ± 0.7	50.7 ± 1.2
	– years	0.623 ± 0.007	7.4 ± 0.3	46.1 ± 0.7	65.0 ± 0.2	61.1 ± 0.7
	– dates	0.637 ± 0.008	7.4 ± 0.5	46.7 ± 0.5	64.2 ± 0.8	60.5 ± 1.6
	– numbers	0.614 ± 0.007	8.0 ± 0.3	46.3 ± 0.7	64.2 ± 1.0	63.1 ± 0.7
	LSTM	0.774 ± 0.006	1.8 ± 0.3	50.4 ± 3.6	77.3 ± 1.4	39.9 ± 1.6
	– years	0.733 ± 0.007	1.3 ± 0.5	41.0 ± 1.2	68.5 ± 0.9	48.3 ± 1.4
	– dates	0.732 ± 0.004	1.4 ± 0.2	42.5 ± 0.6	70.1 ± 1.0	47.7 ± 0.6
	– numbers	0.723 ± 0.007	1.3 ± 0.3	40.1 ± 1.2	68.8 ± 0.7	49.1 ± 1.0

Table 5: Ablation study for BOE and LSTM models. We include a 95% confidence interval of the mean of each metric computed over 6 runs. Study A was conducted only on datapoints from the WOTD dataset with contextual information, and we report the impact of removing the event text from both models. Study B was conducted on all datapoints from the WOTD dataset, and we report the impact of removing tokens denoting years, dates and numbers. Tokens of these types were removed from the event text and from the contextual information. KT = Kendall's Tau, EM = Exact Match, 20Y = Distance under 20Y, 50Y = Distance under 50Y, MAE = Mean Absolute Error.