Challenging America: Modeling language in longer time scales

Anonymous ACL submission

Abstract

The aim of the paper is to apply, for historical texts, the methodology used commonly to solve various NLP tasks defined for contemporary data, i.e. pre-train and fine-tune large Transformer models. This paper introduces an ML challenge, named Challenging America (ChallAm), based on OCR-ed excerpts from historical newspapers collected from the Chronicling America portal. ChallAm provides a dataset of clippings, labeled with metadata on their origin, and paired with their textual contents retrieved by an OCR tool. Three, publicly available, ML tasks are defined in the challenge: to determine the article date, to detect the location of the issue, and to deduce a word in a text gap (cloze test). Strong baselines are provided for all three ChallAm tasks. In particular, we pre-trained a RoBERTa model from scratch from the historical texts. We also discuss the issues of discrimination and hate-speech present in the historical American texts.

1 Introduction

The dominant approach in the design of current NLP solutions consists in (pre-)training a large neural language model, usually applying a Transformer architecture, such as GPT-2, RoBERTa or T5, and fine-tuning the model for specific tasks (Devlin et al., 2018; Raffel et al., 2019). The solutions are evaluated on benchmarks such as GLUE ((Wang et al., 2018)) or SuperGLUE ((Wang et al., 2019)), which allow comparing the performance of various methods designed for the same purpose. A main feature of a good NLP benchmark is the clear separation between train and test sets. This requirement prevents data contamination, when the model (pre-)trained on huge data might have "seen" the test set.

The expansion of digital information is proceeding in two directions on the temporal axis. In the forward direction, new data are made publicly available on the Internet every second. What is less obvious is that, in the backward direction, older and older historical documents are digitized and disseminated publicly.

To the best of our knowledge, our paper introduces the first benchmark which serves to use and evaluate the "pre-train and fine-tune scenario" applied to a massive collection of historical texts.

The very idea of building language models on historical data is not new. The Google Ngram Viewer (Michel et al., 2011) is based on large amounts of texts from digitized books. The corpus as a whole is not open for the NLP community – only raw n-gram statistics are available. The temporal information is crude (at best, the year of publication is given) and the corpus is heterogeneous (in fact, it is a dump of digitized books of any origin).

In our research, we use one of the richest sources of homogeneous historical documents, **Chronicling America**, a collection of digitized newspapers that cover the publication period of over 300 years (with significant coverage of 150 years), and design an NLP benchmark that may open new opportunities for the modeling of the historical language.

Recently, time-aware language models such as Temporal T5 (Dhingra et al., 2021) and TempoBERT (Rosin et al., 2021) have been proposed. They focus on modern texts dated yearly, whereas we extend language modeling towards both longer time scales and more fine-grained (daily) resolution, using massive amounts of historical texts.

The contribution of this paper is as follows:

- We extracted a large corpus of English historical texts that may serve to pre-train historical language models (Section 5).

  These are the main features of the corpus:

  - the corpus size is 201 GB, which is comparable with contemporary text data for

training massive language models, such as GPT-2, RoBERTa or T5;
  - the corpus is free of spam and noisy data (although the quality of OCR processing varies);
  - texts are dated with a daily resolution, hence a new dimension of time (on a fine-grained level) can be introduced into language modeling;
  - the whole corpus is made publicly available;

- Based on selected excerpts from Chronicling America, we define a suite of challenges (named Challanging America, or ChallAm in short) with three ML tasks combining layout recognition, information extraction and semantic inference (Section 7). We hope that ChallAm will give rise to a historical equivalent of the GLUE (Wang et al., 2018) or SuperGLUE (Wang et al., 2019) benchmarks.
  - In particular, we provide a tool for the intrinsic evaluation of language models based on a word-gap task, which calculates the model perplexity in a comparative scenario (the tool may be used in competitive shared-tasks) (Section 7.3).

- We propose a "future-proof" methodology for the creation of NLP challenges: a challenge is automatically updated whenever the underlying corpus is enriched (Section 6.3).

- We introduce a method for data preparation that prevents data contamination (Section 6.3).

- We train base Transformer (RoBERTa) models for historical texts (Section 5). The models are trained on texts spanning 100 years, dated with a daily resolution.

- We provide strong baselines for three ChronAm challenges (Section 8).

- We take under consideration the issue of discrimination and hate speech in the historical American texts. To this end we have applied up-to date methods to filter out the abusive content from the data (Section 9).

2 Chronicling America

In 2005 a partnership between the National Endowment for the Humanities and the Library of Congress launched the National Digital Newspaper Program, to develop a database of digitized documents with easy access. The result of this 15-year effort is Chronicling America – a website[1] which provides access to selected digitized newspapers, published from 1690 to the present. The collection includes approximately 140 000 bibliographic title entries and 600 000 library holdings records, converted to the MARCXML format. The portal supports an API which allows accessing of the data in various ways, such as the JSON format, BulkData (bulk access to data) or Linked Data,[2] or searching of the database with the OpenSearch protocol.[3]. The accessibility of data in various forms makes Chronicling America a valuable source for the creation of datasets and benchmarks.

The portal serves as a resource for various research activities. Cultural historians may track performances and events of their interest in a resource which is easily and openly accessible, as opposed to commercial databases or "relatively small collections of cultural heritage organizations whose online resources are isolated and difficult to search" (Clark, 2014). The database enables searching for the first historical usages of word terms. For instance, thanks to the Chronicling America portal, it was discovered in (Cibaroğlu, 2019) that the term "fake news" was first used in 1889 in the Polish newspaper *Ameryka*.

The resource is helpful in research aiming to improve the output of the OCR process. The authors of (Nguyen et al., 2019) study OCR errors occurring in several digital databases – including Chronicling America – and compare them with human-generated misspellings. The research results in several suggestions for the design of OCR post-processing methods. The implementation of an unsupervised approach in the correction of OCR documents is described in (Dong and Smith, 2018). Two million issues from the Chronicling America collection of historic U.S. newspapers are used in a sequence-to-sequence model with attention.

Chronicling America is a type of digitized resource that may be of wide use for both humanities and computational research. We prepared datasets and challenges based on the data from the Chronicling America resource. We hope that our initiative will bring about research that will facilitate the

[1] https://chroniclingamerica.loc.gov
[2] https://www.w3.org/standards/semanticweb/data
[3] https://opensearch.org/

development of ML-based processing tools, and consequently increase access to digitized resources for the humanities.

An example of an ML tool based on Chronicling America is described in (Lee et al., 2020). The task consisted in predicting bounding boxes around various types of visual content: photographs, illustrations, comics, editorial cartoons, maps, headlines and advertisements. The training set was crowdsourced and included over 48K bounding boxes for seven classes. Using a pre-trained Faster-RCNN detection object, the researchers achieved an average accuracy of 63.4%. Both the training set and the model weights file are publicly available. Still, it is difficult to estimate the value of the results achieved without any comparison with other models trained on the same data.

In our proposal we go a step further. We provide and make available training data from Chronicling America for three ML tasks. For each task we develop and share baseline solutions. Alternative solutions can be submitted to an evaluation platform to be evaluated automatically and compares against the baselines.

3 Similar Machine Learning datasets and challenges

This section concerns ML challenges which deliver labeled OCR documents as training data, a definition of the processing task, and an evaluation environment to estimate the performance of uploaded solutions. More often than not, such challenges concern either layout recognition (localization of layout elements) or Key Information Extraction (finding, in a document, precisely specified business-actionable pieces of information). Layout recognition in Japanese historical texts is described in (Shen et al., 2020). The authors use deep learning-based approaches to detect seven types of layout element categories: Page Frame, Text Region, Text Row, Title Region, etc. Some Key Information Extraction tasks are presented in (Stanisławek et al., 2021). The two datasets described there contain, respectively, NDA documents and financial reports from charity organizations. The tasks for the datasets consist in detecting data points, such as effective dates, interested parties, charity address, income, spending. The authors provide several baseline solutions for the two tasks, which apply up-to-date methods, pointing out that there is still room for improvement in the KIE research area. A challenge that comprises both layout recognition and KIE is presented in (Huang et al., 2019) – the challenge is opened for the recognition of OCR-scanned receipts. In this competition (named ICDAR2019) three tasks are set up: Scanned Receipt Text Localization, Scanned Receipt OCR, and Key Information Extraction from Scanned Receipts.

A common feature of the above-mentioned challenges is the goal of retrieving information that is explicit in the data (a text fragment or layout coordinates). Our tasks in ChallAm go a step further: the goal is to infer the information from the OCR image rather than just retrieve it.

Similar challenges for two out of the three tasks introduced in this paper have been proposed before for the Polish language:

- a challenge for temporal identification (Graliński and Wierzchoń, 2018); the challenge was based on a set of texts coming from Polish digital libraries, dated between the years 1814 and 2013;
- a challenge for "filling the gap" (RetroGap) (Graliński, 2017) with the same training set as above.

The training sets for those challenges were purely textual. Here, we introduce the challenges with the addition of original images (clippings), though we do not use graphical features in baselines yet.

4 Data processing

The PDF files were downloaded from Chronicling America and processed using a pipeline primarily developed for extracting texts from Polish digital libraries (Graliński, 2013, 2019). Firstly, the metadata (including URL addresses for PDF files) were extracted by a custom web crawler and then normalized; for instance, titles were normalized using regular expressions (e.g. *The Bismarck tribune. [volume], May 31, 1921* was normalized to *THE BISMARCK TRIBUNE*). Secondly, the PDF files were downloaded and the English texts were processed into DjVu files (as this is the target format for the pipeline) using the pdf2dvju tool[4]. The original OCR text layer was retained (the files were not re-OCRed, even though, in some cases, the quality of OCR was low).

[4]http://jwilk.net/software/pdf2djvu

Table 1: Statistics for the raw data obtained from the Chronicling America website

Documents with metadata obtained	1 877 363
. . . in English	1 705 008
. . . downloaded	1 683 836
. . . processed into DjVu files	1 665 093

Table 1 shows a summary of the data obtained at each processing step. Two factors were responsible for the fact that not 100% of files were retained at each phase: (1) issues in the processing procedures (e.g. download failures due to random network problems or errors in the PDF-to-DjVu procedure that might be handled later); (2) some files are simply yet to be finally processed in the ongoing procedure.

The procedure is executed in a continuous manner to allow the future processing of new files that are yet to be digitized and made public by the Chronicling America initiative. This solution requires a *future-proof* procedure for splitting and preparing data for machine-learning challenges. For instance, the assignment of documents to the training, development and test sets should not change when the raw data set is expanded. Such a procedure is described in Section 6.

5 Data for unsupervised training

The state of the art in most NLP tasks is obtained by training a neural-network language model on a large collection of texts in an unsupervised manner and fine-tuning the model on a given downstream task. At present, the most popular architectures for language models are Transformer (Devlin et al., 2018) models (earlier, e.g. Word2vec (Mikolov et al., 2013) or LSTM models (Peters et al., 2017)). The data on which such models are trained are almost always modern Internet texts. The high volume of texts available at Chronicling America, on the other hand, makes it possible to train large Transformer models for historical texts.

Using a pre-trained language model on a downstream task bears the risk of *data contamination* – the model might have been trained on the task test set and this might give it an unfair edge (see (Brown et al., 2020) for a study of data contamination in the case of the GPT-3 model when used for popular English NLP test sets). This issue should be taken into account from the very beginning. In our case, we release a dump of all Chronicling America texts (for pre-training language models), but limited only to the 50% of texts that would be assigned to the training set (according to the MD5 hash). This dump contains *all* the texts, not just the excerpts described in Section 6.2. As the size of the dump is 74.0G characters, it is on par with the text material used to train, for instance, the GPT-2 model.

We also release a RoBERTa Base ChallAm model trained on the text corpus. The model was trained from scratch, i.e. it was *not* based on the weights of the original RoBERTa model (Liu et al., 2019). The BPE dictionary was also induced anew.

Two versions of the RoBERTa ChallAm model were prepared: one was trained with temporal metadata encoded as a prefix of the form `year: YYYY, month: MM, day: DD, weekday: WD`, another, for comparison, without such a prefix. The ChallAm models have the same numbers of parameters as the original RoBERTa Base (125M). Each model was trained on two Tesla V100 32GB GPUs for 9 days.

6 Procedure for preparing challenges

We created a pipeline that can generate various machine learning challenges. The pipeline input should consist of DjVu image files, text (OCR image), and metadata. Our main goals are to keep a clear distinction between dataset splits and to assure the reproducibility of the pipeline. This allows potential improvement to current challenges and the generation of new challenges without dataset leaks in the future. We achieved this by employing *stable* pseudo-randomness by calculating an MD5 hash on a given ID and taking the modulo remainder from integers from certain preset intervals. These pseudo-random assignments are not dependent on any library, platform, or programming language (using a fixed seed for the pseudo-random generator might not give the same guarantees as using MD5 hashes), so they are easy to reproduce.

This procedure is crucial to make sure that challenges are *future-proof*, i.e.:

- when the challenges are re-generated on the same Chronicling America files, exactly the same results are obtained (including text and image excerpts; see Section 6.2);
- when the challenges are re-generated on a larger set of files (e.g. when new files are digitized for the Chronicling America project),

the assignments of existing items to the train/dev/test sets will not change.

6.1 Dataset structure

All three of our machine learning challenges consist of training (train), development (dev), and test sets. Each document in each set consists of excerpts from a newspaper edition. One newspaper edition provides a maximum of one excerpt. Excerpts in the datasets are available as both a cropped PNG file from the newspaper scan (a "clipping") and its OCR text. This makes it possible to employ image features in machine learning models (e.g. font features, paper quality). A solution might even disregard the existing OCR text layer and re-OCR the clipping or just employ an end-to-end model. (The OCR layer is given as it is, with no manual correction done – this is to simulate realistic conditions in which a downstream task is to be performed without a perfect text layer.)

Sometimes additional metadata are given. For the train and dev datasets, we provide the expected data. For the test dataset, the expected data are not released. These data are used by the evaluation platform during submission evaluation. All newspaper and edition IDs are encoded to prevent participants from checking the newspaper edition in the Chronicling America database. The train and dev data may consist of all documents which meet our criteria for text excerpts, so the data may be unbalanced with respect to publishing years and locations. We tried to balance the test sets as regards the years of publication (the year-prediction and word-gap challenges) or locations (the geo-prediction challenge), though it is not always possible due to large imbalances in the original material.

6.2 Selecting text excerpts

The details of the procedure for selection of text excerpts is given in Appendix A. A sample excerpt is shown in Figure 1a. Note that excerpts are selected using a stable pseudo-random procedure based on the newspaper edition ID (similarly to the way the train/dev/test split is done, see Section 6.3).

6.3 Train/dev/test split

Each newspaper has its newspaper ID (i.e. normalized title, as described in Section), and each newspaper edition has its newspaper edition ID. We separate newspapers within datasets, so for instance, if one newspaper edition is assigned to the dev set, all editions of that newspaper are assigned to the dev set. All challenges share common train and dev datasets and no challenges share the same test set. This prevents one from checking expected data from other challenges. The set splits are as follows: 50% for train, 10% for dev, 5% for each challenge test set. This makes it possible to generate eight challenges with different test sets. In other words, there is room for another five challenges in the future (again this is consistent with the "future-proof" principle of the whole endeavor).

7 Challenging America tasks

In this section, we describe the three tasks defined in the challenge. They are released on an evaluation platform, which enables the calculation of metrics both offline and online, as well as the submission of solutions. An example of text from an excerpt given in those tasks is shown in Figure 1b.

7.1 RetroTemp

This is a temporal classification task. Given a normalized newspaper title and a text excerpt, the task is to predict the publishing date. The date should be given in fractional year format (e.g. 1 June 1918 is represented as the number 1918.4137, and 31 December 1870 as 1870.9973).

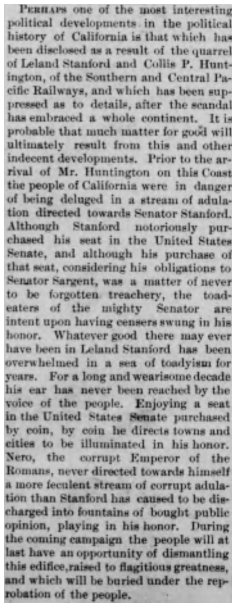
Hence, solutions to the challenge should predict the publication date with the greatest precision possible (i.e. day if possible). The fractional format will make it easy to accommodate even more precise timestamps, for example, if modern Internet texts (e.g. tweets) are to be added to the dataset.

Due to the regression nature of the problem, the evaluation metric is RMSE (root mean square error).

The motivation behind the RetroTemp challenge is to design tools that may help supplement the missing metadata for historical texts (the older the document, the more often it is not labeled with a time stamp). Even if all documents in a collection are time-stamped, such tools may be useful for finding errors and anomalies in metadata.

7.2 RetroGeo

The task is to predict the place where the newspaper was published, given a normalized newspaper title, text excerpt, and publishing date in fractional year format. The expected format is a latitude and longitude. In the evaluation the distance on the sphere between output and expected data is calculated using the haversine formula, and the mean



PERHAPS one of the most interesting political developments in the political history of California is that which has been disclosed as a result of the quarrel of Leland Stanford and Collis P. Huntington, of the Southern and Central Pacific Railways, and which has been suppressed as to details, after the scandal has embraced a whole continent. It is probable that much matter for good will ultimately result from this and other indecent developments. Prior to the arrival of Mr. Huntington on this Coast the people of California were in danger of being deluged in a stream of adulation directed towards Senator Stanford. Although Stanford notoriously purchased his seat in the United States Senate, and although his purchase of that seat, considering his obligations to Senator Sargent, was a matter of never to be forgotten treachery, the toad-eaters of the mighty Senator are intent upon having censers swung in his honor. Whatever good there may ever have been in Leland Stanford has been overwhelmed in a sea of toadyism for years. For a long and wearisome decade his ear has never been reached by the voice of the people. Enjoying a seat in the United States Senate purchased by coin, by coin he directs towns and cities to be illuminated in his honor. Nero, the corrupt Emperor of the Romans, never directed towards himself a more feculent stream of corrupt adulation than Stanford has caused to be discharged into fountains of bought public opinion, playing in his honor. During the coming campaign the people will at last have an opportunity of dismantling this edifice, raised to flagitious greatness, and which will be buried under the reprobation of the people.

(a) An excerpt.

```
Perhaps one of the most interesting political developments
in tbe political history of California is that which has been
disclosed as a result of the quarrel of Leland Stanford and
Collis P. Hunt- ington, of the Southern and Central Pa- cific
Railways, and which has been sup- pressed as to details, after
the scandal has embraced a whole continent.  It is probable
that much matter for good will ultimately result from this
and other indecent developments.  Prior to the ar- rival of
Mr.  Huntington on this Coast the people of California were
in danger of being deluged in a stream of adula- tion directed
towards Senator Stanford.  Although Stanford notoriously pur-
chased his seat in the United States Senate, and although bis
purchase of that seat, considering his obligations to Senator
Sargent, was a matter of never to be forgottoa treachery, the
toad- eaters of the might}' Senator are intent upon having
censers swung in his ...
```

(b) Fragment of a text from an excerpt.

Figure 1: An example of an excerpt

value of errors is reported.

The motivation for the task (besides the supplementation of missing or wrong data) is to allow research on news propagation. Even if a news article is labeled with the localization of its issue, an automatic tool may infer that it was originally published somewhere else.

7.3 RetroGap

This is a task for language modeling. The middle word of an excerpt is removed in the input document (in both text and image), and the task is to predict the removed word, given the normalized newspaper title, the text excerpt, and the publishing date in fractional year format (in other words, it is a cloze task). The output should contain a probability distribution for the removed word (not just a word or a single probability). The metric is perplexity; PerplexityHashed, to be precise, as implemented in the GEval evaluation tool (Graliński et al., 2019), the modification is analogous to LogLossHashed in (Graliński, 2017), its goal is to ensure proper evaluation in the competitive (shared-task) setup (i.e. avoid self-reported probabilities and ensure objective comparison of all reported solutions, including out-of-vocabulary words).

7.4 Statistics

The data consists of the text excerpts written between the years 1798 and 1963. The mean publication year of the text excerpts is 1891. Excerpts between the years 1833 and 1925 make up about 96% of the data in the train set (cf. Figure 2a), but only 85% in the dev and test sets, which are more uniform (due to balancing described in Section 6.3, cf. Figure 2c). There are 432 000 excerpts in the train set, 10 500 in the dev set and 8 500 in the test set. These numbers are consistent across the challenges. The average excerpt length is 1 745 characters with 323.8 words, each one containing from 150 words up to 583 words.

The length of each text in the excerpts seems to have a negative correlation with publication date – the later the text was published, the shorter snippet text (on average) it contains (see Figure 2b and 2d).

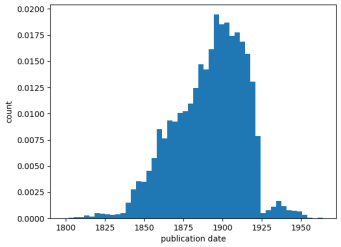
8 Baselines

Baselines for all three tasks are available at the evaluation platform.[5] The baselines (see Tables 2 and 3) include, for each model, its score in the appropriate metric as well as the Git SHA1 reference code (in curly brackets).[6]

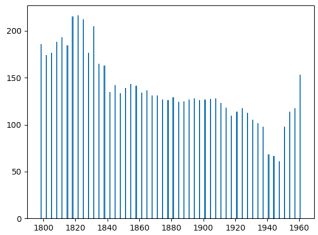
We distinguish between self-contained submissions, which use only data provided in the task, and non-self-contained submissions, which use external data, e.g. publicly available pre-trained transform-

[5] To be revealed after review

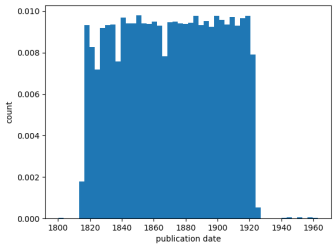
[6] The outputs and some of the scripts used are available in supplementary materials, later to be revealed at the evaluation platform.



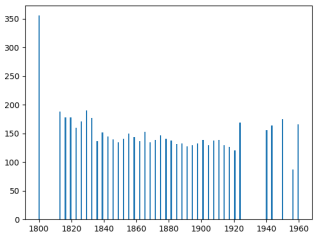
(a) Excerpt counts vs. publication dates in train set.



(b) Average excerpt length vs. publication dates in train set.



(c) Excerpt counts vs. publication dates in dev/test set.



(d) Average excerpt length vs. publication dates in dev/test set.

Figure 2: Statistics for the RetroTemp challenge

ers. Our baselines take into account only textual features.

More detailed analysis of the baseline performance is given in Appendix C. The current top performing models have the most difficulty with texts which (1) are older, (2) contain OCR noise, (3) come from less popular locations (especially, in the west).

8.1 RetroTemp and RetroGeo

The baseline solutions for RetroTemp and RetroGeo were prepared similarly. RetroGeo requires two values (latitude and longitude) – we treat them separately and train two separate models for them.

For the self-contained models we provide the mean value from the train test, the linear regression based on TF-IDF and the BiLSTM (bidirectional long short-term memory) method.

For non-self-contained submissions, we incorporate RoBERTa (Liu et al., 2019) models released in two versions: base (125M params) and large (355M params). The output features are averaged, and the linear layer is added on top of this. Both RoBERTa and the linear layer were fine-tuned during training.

The best self-contained models are BiLSTM submissions in both tasks. Non-self-contained submissions result in much higher scores than self-contained models. In both tasks, RoBERTa-large with linear layer provides better results than RoBERTa-base.

For the RetroTemp challenge we also provide results obtained with the RoBERTa model pretrained from scratch (see Section 5). Even though the model without time-related prefix was used, the results are significantly better than the original RoBERTa Base: the confidence intervals obtained with bootstrap sampling are, respectively, 10.81±0.21 and 12.10±0.22 (single runs are reported).

Hyperparameter setup is described in Appendix B.

8.2 RetroGap

For non-self-contained submissions, we applied RoBERTa in base and large version without any fine-tuning. Since standard RoBERTa training does not incorporate any data, but text, we didn't include temporal metadata during inference.

For self-contained submissions, we applied RoBERTa Challam base both in version with a date and without a date.

RoBERTa ChallAm base with date is better than RoBERTa ChallAm base without date. This means the incorporation of temporal metadata has a positive impact on MLM task. Both self-contained submissions are better than the standard RoBERTa base, so our models trained on historical data per-

Table 2: Baseline results for the RetroTemp/Geo challenges. * indicates non-self-contained models.

Model	RetroTemp		RetroGeo	
	git ref	RMSE	git ref	Haversine
mean from train	{fbf19b}	31.50	{766824}	1321.47
tf-idf with linear regression	{63c8d4}	17.11	{8acd61}	2199.36
BiLSTM	{f7d7ed}	13.95	{d3d376}	972.71
RoBERTa Base + linear layer*	{1159e6}	12.07	{08412c}	827.13
RoBERTa Large + linear layer*	{2e79c8}	**8.15**	{7a21dc}	**651.20**
RoBERTa ChallAm Base + linear layer*	{d0ddf4}	10.80	—	

Table 3: Baseline results for the RetroGap challenge. * indicates non-self-contained models.

Model	git ref	Perplexity
RoBERTa base (no fine-tune)	{166e03}	72.10
RoBERTa large (no fine-tune)	{bf5171}	**52.58**
RoBERTa ChallAm Base (without date)*	{f96da0}	56.64
RoBERTa ChallAm Base (with date)*	{3ebfc0}	**53.76**

forms better than model trained on regular data if the same base model size is considered. Since we didn't train RoBERTa ChallAm large, we can't confirm this holds true, when it comes to large RoBERTa models. The standard RoBERTa large is the best performing model, so in this case, a larger model is better even if not trained on the data from different domain.

9 Ethical issues

We share the data from Chronicling America, following the statement of the Library of Congress: "The Library of Congress believes that the newspapers in Chronicling America are in the public domain or have no known copyright restrictions."[7]

Historical texts from American newspapers may be discriminatory, either explicitly or implicitly, particularly regarding race and gender. Recent years have seen research on the detection of discriminatory texts. In (Xia et al., 2020) adversarial training is used to mitigate racial bias. In (Field and Tsvetkov, 2020) the authors "take an unsupervised approach to identifying gender bias against women at a comment level and present a model that can surface text likely to contain bias." The most recent experiments on the topic ((Caselli et al., 2021), (Aluru et al., 2020)) result in re-trained BERT models for abusive language detection in English. We use one of them, DeHateBERT (Aluru et al., 2020), to filter out the abusive texts in the ChallAm dataset. We filtered out items that either (1) are marked as abusive speech by DeHateBERT with the probability greater than 0.75 or (2) contain words from a list of blocked words. The fraction of filtered out texts was 2.04-2.40% (depending on the challenge and set).

[7] https://chroniclingamerica.loc.gov/about

10 Conclusions

This paper has introduced a challenge based on OCR excerpts from the Chronicling America portal. The challenge consists of three tasks: guessing the publication date, guessing the publication location, and filling a gap with a word. We propose baseline solutions for all three tasks.

Chronicling America is an ongoing project, as we define our challenge in such a way that it can easily evolve in parallel with the development of Chronicling America. Firstly, any new materials appearing on the portal can be automatically incorporated into our challenge. Secondly, the challenge is open for five yet undefined ML tasks.

References

Sai Saket Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models

for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in english.

Mehmet Cibaroğlu. 2019. Post-truth in social media. 6:87–99.

Maribeth Clark. 2014. A survey of online digital newspaper and genealogy archives: Resources, cost, and access. *Journal of the Society for American Music*, 8:277–283.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2021. Time-aware language models as temporal knowledge bases.

Rui Dong and David Smith. 2018. Multi-input attention for unsupervised OCR correction. pages 2363–2372.

Anjalie Field and Yulia Tsvetkov. 2020. Unsupervised discovery of implicit gender bias. *CoRR*, abs/2004.08361.

Filip Graliński and Piotr Wierzchoń. 2018. RetroC—A Corpus for Evaluating Temporal Classifiers. In *Human Language Technology. Challenges for Computer Science and Linguistics. 7th Language and Technology Conference, LTC 2015*, pages 101–111. Springer.

Filip Graliński, Anna Wróblewska, Tomasz Stanisławek, Kamil Grabowski, and Tomasz Górecki. 2019. GEval: Tool for debugging NLP datasets and models. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 254–262, Florence, Italy. Association for Computational Linguistics.

Filip Graliński. 2013. Polish digital libraries as a text corpus. In *Proceedings of 6th Language & Technology Conference*, pages 509–513, Poznań. Fundacja Uniwersytetu im. Adama Mickiewicza.

Filip Graliński. 2017. (Temporal) language models as a competitive challenge. In *Proceedings of the 8th Language & Technology Conference*, pages 141–146. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu.

Filip Graliński. 2019. *Against the Arrow of Time. Theory and Practice of Mining Massive Corpora of Polish Historical Texts for Linguistic and Historical Research*. Wydawnictwo Naukowe UAM, Poznań.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. 2019. ICDAR2019 competition on scanned receipt OCR and information extraction. *2019 International Conference on Document Analysis and Recognition (ICDAR)*.

Benjamin Lee, Jaime Mears, Eileen Jakeway, Meghan Ferriter, Chris Adams, Nathan Yarasavage, Deborah Thomas, Kate Zwaard, and Daniel Weld. 2020. The newspaper navigator dataset: Extracting and analyzing visual content from 16 million historic newspaper pages in Chronicling America.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Thi-Tuyet-Hai Nguyen, Adam Jatowt, Mickael Coustaty, Nhu-Van Nguyen, and Antoine Doucet. 2019. Deep statistical analysis of OCR errors for effective post-OCR processing. In *Proceedings of the 18th Joint Conference on Digital Libraries*, JCDL '19, page 29–38. IEEE Press.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*.

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

Guy D. Rosin, Ido Guy, and Kira Radinsky. 2021. Time masking for temporal language models.

Zejiang Shen, Kaixuan Zhang, and Melissa Dell. 2020. A large dataset of historical Japanese documents with complex layouts. *CoRR*, abs/2004.08686.

Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. Kleister: Key information extraction datasets involving long documents with complex layouts. In *Document Analysis and Recognition – ICDAR 2021*, pages 564–579, Cham. Springer International Publishing.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. 2020. Demoting racial bias in hate speech detection. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*, pages 7–14, Online. Association for Computational Linguistics.

A Procedure for selecting text excerpts

The OCR text follows the newspaper layout, which is defined by the following entities: page, column, line. Each entity has x_0, y_0, x_1, y_1 coordinates of text in the DjVu document. Still, various errors may occur in the OCR newspaper layout (e.g. two columns may be split into one). We intend to select only excerpts which preserve the correct output. To this end, we select only excerpts that fulfill the following conditions:

1. There are between 150 and 600 text tokens in the excerpt. The tokens are words separated by whitespaces.
2. The y coordinates of each line are below the y coordinates of the previous line.
3. The x_0 coordinate of each line does not differ by more than 15% from the x_0 coordinate of the previous line.
4. The x_1 coordinate is not shifted to the right more than 15% from the x_1 coordinate of the previous line.

If the newspaper edition contains no such excerpts, we reject it. If there is more than one such excerpt, we select one excerpt using a stable pseudo-random procedure based on the newspaper edition ID.

This procedure produces text excerpts with images consisting of OCR texts only. The excerpts are downsized to reduce the size to an appropriate degree to maintain good quality. We do not pre-process images in any other way, so excerpts may have different sizes, height-to-width ratios, and colors.

B Hyperparameter setup

Hyperparameters were determined on the development set, training on a limited number of examples. In particular, for fine-tuning RoBERTa models the following hyperparameters were used:

- optimizer: AdamW
- learning rate: 0.000001
- batch size: 4
- early-stopping patience: 3
- warm-up steps: 10000

C Analysis of the best baselines

See Table 4 and 5 for the list of top 30 features correlating most with, respectively, the worst and bad results in ChallAm challenges (as returned by the GEval tool with the option `-worst-features -numerical-features` (Graliński et al., 2019)). The features are tokens within the input (`in:`), expected output (`exp:`) and the actual output (`out:`), or numerical features such as high/low value (`:=+`/`:=-`) or length/shortness of a text (`:+#`/`:-#`).

As can be seen the bottleneck for the current best model is due to:

- old texts (`:=-` in RetroTemp),
- OCR noise (cf. short words such *ni*, *ol*, *j* or punctuation marks likely to be introduced by OCR misrecognitions),
- less popular publication locations (especially far west).

Obviously, year references (*1902*, *1904*) make it easy to guess the publication texts (in RetroTemp), whereas in RetroGap some non-content words such as *the*, *and*, *of* are easy to guess for the language model (even if their garbaged form, e.g. *ot*, *ol*, needs to be accounted for in the probability distribution).

Table 4: Features highly correlating with bad results

RetroTemp	RetroGeo	RetroGap
exp:=-	exp:=#+	exp:=#+
in<Text>:;	in<Text>:=+	exp:,
in<Text>:nold	exp:-100.445882	exp:.
in<Text>:ni	exp:39.78373	out:.
in<Text>:she	exp:-115.763123	out:-
out:=-	exp:40.832421	in<LeftContext>:n
in<Text>:”	exp:-93.101503	out:,
in<Text>:aim	exp:44.950404	out:;
in<Text>:sav-	exp:-112.730038	out:’
in<Text>:ii	exp:46.395761	out:*
in<Text>:rifle	exp:-97.337545	in<RightContext>:*
in<Text>:hut	exp:37.692236	in<LeftContext>:>
in<Text>:!	exp:-76.062727	out:=#-
in<Text>:guilt	exp:39.697887	in<RightContext>:>
in<Text>:nLeave	exp:-106.487287	in<LeftContext>:i
in<Text>:ol	exp:31.760037	out:!
in<Text>:cold	exp:-81.772437	exp:;
in<Text>:contemplate	exp:24.562557	in<LeftContext>:*
in<Text>:nI	exp:-71.880373	in<RightContext>:l
in<Text>:thee	exp:44.814771	out:"
in<Text>:Ben-	out:=#+	out:|
in<Text>:1945	exp:-135.313889	in<LeftContext>:l
in<Text>:God	exp:59.458333	out:1
in<Text>:it	exp:-112.077346	exp:"
in<Text>:noi	exp:33.448587	in<LeftContext>:<
in<Text>:man’s	exp:-122.330062	in<LeftContext>:-
in<Text>:Roman	exp:47.603832	in<RightContext>:|
in<Text>:I	exp:-112.942369	out:i
in<Text>:Henry	exp:46.128794	out:j
in<Text>:nford	exp:-90.184225	in<LeftContext>:e

Table 5: Features highly correlating with good results

RetroTemp	RetroGeo	RetroGap
in<Text>:Democratic	exp:44.007274	out:Of
in<Text>:defeat	exp:-80.85675	out:The
in<Text>:Secretary	exp:40.900892	out:ana
in<Text>:notice	exp:-77.804161	out:aud
in<Text>:July	exp:39.4301	out:by
in<Text>:General	exp:-79.96021	out:cf
in<Text>:1904	exp:37.274532	out:end
in<Text>:cent	exp:-82.137089	out:for
in<Text>:of	exp:38.844525	out:he
in<Text>:are	exp:-77.859581	out:in
in<Text>:will	exp:39.289184	out:io
in<Text>:1902	exp:-80.344534	out:lo
in<Text>:against	exp:39.280645	out:mat
in<Text>:nbeen	exp:-81.929558	out:of
in<Text>:Minnesota	exp:33.789577	out:ol
in<Text>:1903	exp:-77.321601	out:or
in<Text>:Judicial	exp:37.506699	out:ot
in<Text>:President	exp:-73.986614	out:tc
in<Text>:June	exp:-77.036646	out:te
in<Text>:to	exp:-77.047023	out:th
in<Text>:for	exp:-77.090248	out:tha
in<Text>:hereby	exp:-77.43428	out:that
in<Text>:States	exp:-80.720915	out:the
in<Text>:United	exp:37.538509	out:this
in<Text>:nLouisiana	exp:38.80511	out:tho
in<Text>:county	exp:38.81476	out:tie
in<Text>:State	exp:38.894955	out:tile
in<Text>:Is	exp:40.063962	out:to
in<Text>:cash	exp:40.730646	out:tu
in<Text>:In	out:-158.09514	out:und