Model Details

`American Stories` deploys a modular pipeline to digitize historical newspapers. This section provides details for each component of the pipeline.

Layout Detection

To detect articles, headlines, ads, and other content regions in a newspaper scan, we deploy YOLOv8 (Medium) [18], initialized from the officially released YOLOv8m pretrained checkpoint. We train for 100 epochs on 2,202 labeled newspaper scans with 48,874 total layout objects, using default YOLOv8 hyperparameters except: {`imgsz: 1280, iou: 0.2, max_det: 500`}. The final model achieved a 0.91 mAP50:95 on article bounding boxes and a 0.84 mAP50:95 on headline bounding boxes. We decreased the confidence threshold to 0.1 to increase article and headline recall.

Legibility Classification

Text image bounding boxes are classified as legible, borderline, or illegible, using MobileNetV3 (Small) [5] initialized from the PyTorch Image Models ("timm") [19] pretrained checkpoint. We train for 50 epochs on 979 labeled article, headline, and image caption examples. 678 of the labeled examples were legible, 192 borderline, and 109 illegible. The model was trained with weighted Cross Entropy Loss: weights [2.0, 1.0, 1.0] for legible, borderline, and illegible classes, respectively. The following specifications were used: {`resolution: 256, learning rate: 2e-3`}. The learning rate was multiplied by 0.1 every twenty epochs.

Text Line Detection

Line bounding boxes are detected using YOLOv8 (Small) [18] initialized from the official YOLOv8s pretrained checkpoint. We train first for 100 epochs on 4000 synthetically generated articles, with default YOLOv8 hyperparameters. After synthetic training, the model was additionally trained for 50 epochs on 373 hand-annotated article and headline crops, with default YOLOv8 hyperparameters except for the following: {`resolution: 640, initial learning rate: 0.02, final learning rate: 0.002`}.

Word and Character Localization

Words and characters are detected using YOLOv8 (Small) [18], initialized from the official YOLOv8s pretrained checkpoint. We train first for 100 epochs on 8000 synthetically generated textlines with default YOLOv8 hyperparameters. After synthetic training, the model was additionally trained for 100 epochs on 684 hand-annotated text line images, with default hyperparameters except for the following: {`resolution: 640, initial learning rate: 0.02, final learning rate: 0.001`}. Each hand-annotated line image was replicated three times with random augmentations along three axes: random rotation between -1°and 1°, random image brightness shift from -30 to 30%, and randomly applied blur at the 0-4px level. On average, text line examples contained 4.3 words and 23.6 characters.

Word Recognition

Building upon the architecture in [2], we train word recognition as a nearest neighbor image retrieval problem. As described in the main text, the training dataset for the model consists of digital renders of words created using 43 fonts, silver quality data from the target dataset created by applying the EffOCR-C (Small) model from [2] to a random sample of days, and a small number of randomly selected hand labeled word crops. We limited the number of crops with model-generated labels to 20 - so each word can have 0-20 silver-quality crops depending upon its frequency of occurrence in our random sample. This limit is binding for common words, *e.g.,* "the".

The recognizer is trained using the Supervised Contrastive ("SupCon") loss function [7], a generalization of the InfoNCE loss [11] that allows for multiple positive and negative pairs for a given anchor. In particular, we work with the "outside" SupCon loss formulation

$$\mathcal{L}_{\text{out}}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_{\text{out},i}^{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau\right)}$$

as implemented in PyTorch Metric Learning [10], where τ is a temperature parameter, i indexes a sample in a "multiviewed" batch (in this case multiple fonts/augmentations of the same word), $P(i)$ is the set of indices of all positives in the multiviewed batch that are distinct from i, $A(i)$ is the set of all indices excluding i, and z is an embedding of a sample in the batch [7].

To create training batches for the recognizer, we use a custom m per class sampling algorithm without replacement, adapted from the PyTorch Metric Learning repository [10]. The m word variants for each class (word) are drawn from both target documents and augmented digital fonts. We select $m = 4$ and the batch size is 1024, meaning 4 styles of each of 256 different words appear in each batch. For training without hard negatives, we define an epoch as letting the model see each word (case-sensitive) exactly $m = 4$ times. Sampling for each class occurs without replacement until all variants are exhausted.

In order to converge faster with limited compute, we also implement offline-hard negative mining, batching similar negatives and their corresponding positive anchors together - thus making the contrasts between the positive and negative pairs within a batch especially informative. To create hard negative sets, we render each word using a reference font (Noto-Serif Regular) and embed it to create a reference index. We find $k = 8$ nearest neighbors for each word using this index and the model trained without hard negatives, which yields sets of 8 words that have a similar appearance when rendered with the reference font. We use only the reference font to create these sets because using crops corresponding to all 43 fonts for each word is computationally costly and creates more hard negative sets than we can use in training. We also use each word crop from the target dataset (both silver quality annotations generated with model predictions and gold quality human-annotated predictions) to create hard negative sets. Hence, the total number of hard-negative sets equals the number of words in our dictionary (generated with the reference font) plus the number of word crops from the newspaper data in the training set.

Each hard negative set contains 8 words, with $m = 4$ views per word, which means we can fit 32 randomly sampled hard negative sets within each batch. An epoch is defined as seeing each hard negative set once. Since the number of synthetic views of an image is much larger than the number

of target newspaper crops, whenever newspaper crops are available we force the m views of a word to contain an equal number of synthetic and target crops.

We use a MobileNetV3 (Small) encoder pre-trained on ImageNet1k sourced from the timm [19] library, more specifically, the model *mobilenetv3_small_050*. We use 0.1 as the temperature for SupCon loss and AdamW as the optimizer with Pytorch [12] defaults for all parameters other than weight decay (5e-4) and learning rate. We used Cosine Annealing with Warm Restarts as the learning rate scheduler with a maximum learning rate of $2e-3$, a minimum learning rate of 0, time to first restart (T_0) as the number of batches in an epoch, and restart factor, T_{mult} of 2 using the implementation provided in Pytorch.

While fonts and newspaper crops for each word act as an augmentation on the skeleton of the word, we also add more image-level transformations to improve generalization. These include Affine transformation (only slight translation and scaling allowed), Random Color Jitter, Random Autocontrast, Random Gaussian Blurring, Random Grayscale, Random Solarize, Random Sharpness, Random Invert, Random Equalize, Random Posterize and Randomly erasing a small number of pixels of the image. Additionally, we pad the word to make the image square while preserving the aspect ratio of the word render. We do not use common augmentations like Random Cropping or Center Cropping, to avoid destroying too much information.

The model trained without hard negatives was trained for 50 epochs and with hard negatives, it was trained for 40 epochs. For selecting the best checkpoint, we use 1-CER (OCR Character Error Rate) as the validation metric on the validation set from [2]. We chose the model that performed best in terms of CER when detecting only words on the validation set. This means that if a word is outside of our dictionary, it is forcefully matched to the nearest neighbor in the dictionary. The best model achieved a CER of 4.9% with word-only recognition.

At inference time, words are recognized by retrieving their nearest neighbor from the offline embedding index created with the reference font, using a Facebook Artificial Intelligence Similarity Search backend [6]. The code to train the model and generate training data, as well as the model checkpoints, are made available on our GitHub repo.[1]

Character Recognition

When the nearest neighbor to an embedded word crop in the offline word embedding index is below a cosine similarity threshold of 0.82, we default to character-level recognition. We use the EffOCR-C (Small) model that is developed in [2] for character recognition.

Content Association

This step associates headlines, bylines, and article bounding boxes. We use rule-based methods that exploit the position of article and byline bounding boxes relative to headlines. Specifically, we associated a headline bounding box with an article bounding box if they overlap vertically by more than 1% of the page width, and the bottom of the headline is no greater than 10% of the page height above the top of the article, and no greater than 2% of the page height below the top of the article. If multiple article bounding boxes satisfy these rules for a given headline, then we take the highest. The same rules are used to associate bylines.

[1] `https://github.com/dell-research-harvard/AmericanStories`.

Pipeline Evaluation

As discussed in the main paper, we evaluate the data processing pipeline in an end-to-end fashion, as well as evaluating individual sections, particularly OCR. Here we provide additional details on those evaluations.

OCR Evaluation

Processing 20 million scans required a cost-effective OCR solution, and downstream tasks require highly accurate OCR. We compared custom, open-source, and commercial OCR solutions by accuracy, speed, and cost to determine our final architecture. Character Error Rate measurements were made on two separate validation datasets:

- **CER [2]** Error rate on a dataset of 64 randomly selected Chronicling America textlines, sampled from the entire collection. Textlines were randomly sampled from random scans, then cropped and transcribed. This dataset and its construction is described in detail in [2].
- **CER Day-Per-Decade** Error rate on a sample of 225 total textlines, sampled from all scans in the Chronicling America collection published on March 1st of years ending in "6," from 1856-1926. Unlike the above sample from [2], this sample is balanced across the time periods the predominate the Chronicling America collection. 25 textlines were sampled randomly from random pages published on each of the days. A selection of textlines from this collection, along with their EffOCR transcriptions, are shown in Figure 1. This dataset is designed to be much more challenging than the first, weighting older, harder to read scans more heavily despite their relative scarcity in the Chronicling America collection.

Comparisons are listed in Table 1. Training procedures for EffOCR-Word are described above. See [2] for training procedures, initialization checkpoints, and additional details on training and evaluating comparison models.

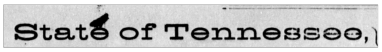
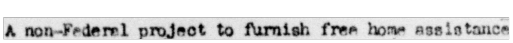
	WASHINGTON, April 1 Ambas-
	FORT WORTH JITNEYS QUIT
	General Plan 5-4-31
	State of Tennessee
	A non-Federal project to furnish free home assistance
	SEED DISTRIBUTION
	Iron, Steel and Tin Workers
	ADVERSE REPORTS ON DEMENT'S NOMINATION.
	IMPROVEMENT IS SHOWN

Figure 1: **Examples of textlines in the Day-Per-Decade evaluation set.** Image crops are shown on the left, with their corresponding EffOCR transcriptions (using the final model set used in the `American Stories` processing pipeline) on the right.

Model/Engine	Seq2Seq?	Transformer?	Pretraining	Parameters	CER [2]	CER Day-Per-Decade	Lines Per Second	Cost Per 10K Lines
EffOCR-C (Base)	×	×	from scratch	112.5 M	0.023	0.062	0.27	$1.77
EffOCR-C (Small)	×	×	from scratch	9.3 M	0.028	0.080	7.28	$0.06
EffOCR-T (Base)	×	✓	from scratch	101.8 M	0.022	0.059	0.17	$2.80
EffOCR-Word (Small)	×	×	**from scratch**	**10.6 M**	**0.015**	**0.043**	**11.60**	**$0.04**
Google Cloud Vision OCR	?	?	off-the-shelf	?	0.005	0.019	?	$15.00
Tesseract OCR (Best)	✓	×	off-the-shelf	1.4 M	0.106	0.170	2.43	$0.19
EasyOCR CRNN	✓	×	off-the-shelf fine-tuned from scratch	3.8 M	0.170 0.036 0.131	0.274 0.157 -	10.75	$0.04
PaddleOCR SVTR	×	×	off-the-shelf fine-tuned from scratch	11 M	0.304 0.103 0.104		7.36	$0.06
TrOCR (Base)	✓	✓	off-the-shelf fine-tuned from scratch	334 M	0.015 0.013 0.809	0.038 0.027 -	0.23	$2.02
TrOCR (Small)	✓	✓	off-the-shelf fine-tuned from scratch	62 M	0.039 0.075 0.773	0.121 0.091 -	0.53	$0.90

Table 1: **Chronicling America Results and Comparisons.** This table reports the performance of different OCR architectures, *off-the-shelf* (without fine-tuning on target data), *fine-tuned* on the Chronicling America training set from initialization on the best public, pre-trained OCR checkpoint, and pre-trained *from scratch* on a consistent, standardized set of synthetic text lines and then fine-tuned on the Chronicling America training set. "?" indicates that the field is unknown due to the proprietary nature of the architecture. Inference speeds are based on an extrapolation from inference speeds measured for EffOCR-Word (Small) to digitize the entire Chronicling America collection using cloud compute.

Of the options we examined, EffOCR-Word (Small) was the clear best option, providing a Character Error Rate under 5% on the hardest evaluation set while offering the cheapest rate per line on an Microsoft Azure Fs4v2 instance.

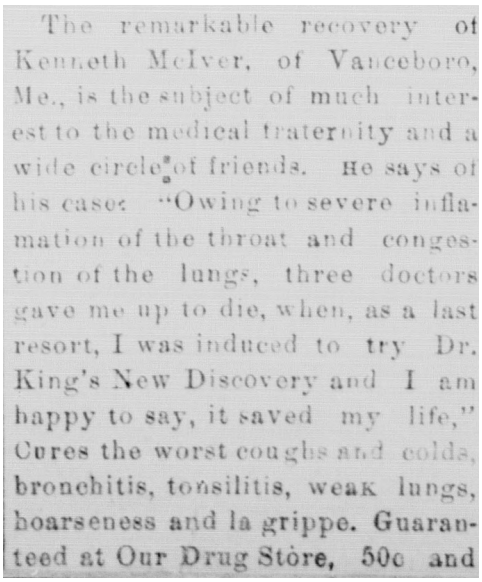
Legibility

Legibility classification was tested on a set of 100 image crops (50 articles and 50 headlines) sampled randomly from the 1,094-image legibility training set. All legibility images were double-entered. Since the goal was to be cautious in classifying images as illegible, where annotators disagreed the more legible of the two labels was used.

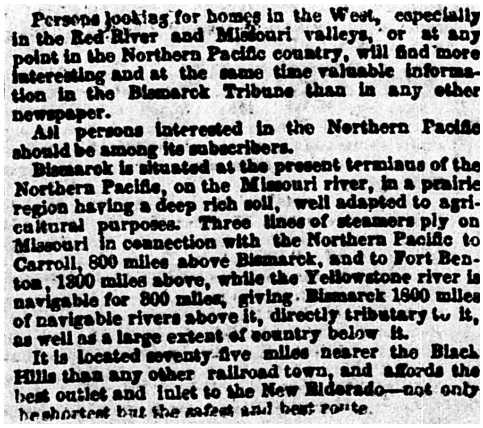
Annotators were instructed to use the following definitions for legibility labeling:

- **Legible**: Greater than 95% of words in an image readable without context from adjacent words.
- **Borderline**: Between 50 and 95% of words in an image readable without context from adjacent words.
- **Illegible**: Less than 50% of words in an image readable without context from adjacent words.

Inter-annotator agreement was 91% between the two annotators. A sample of annotator discrepancies is presented in Figure 2

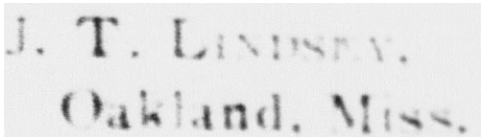


(a) Legible/Borderline



(b) Borderline/Illegible

(c) Legible/Borderline



(d) Borderline/Illegible

Figure 2: **Examples of Inter-Annotator disagreement in legibility labeling.** Images are labeled "Class 1"/"Class 2" to show the two labeled classes. In cases of disagreement, the more legible of the two labels was used for training and evaluation.

Applications

The paper presents multiple applications that can be facilitated by the `American Stories` dataset. This section provides details for each application given.

Topic Classification

To evaluate topic classification, we focused on the topic of politics. As we evaluate at both the scan and article level, for development and test sets we sampled full scans (all articles on the same scan). We took a random sample of up to three front page scans from each election year in our sample. These scans were double-labelled by student research assistants and incongruences were discussed and resolved. We place 20% of these (15 scans, 498 articles) into a development set and the remaining 62 scans (1473 articles) into the test set. Training data was sampled at the article level, rather than the scan level, from the same population of front page articles in election years. Training data was single-labelled by the same research assistants. A sample were double-labelled to check for consistency and they agreed on the labelling in 93% of cases.

To evaluate neural methods, we finetune RoBERTa large [9] on the training set for ten epochs, with a batch size of 16, and a learning rate of 2e-6.

For evaluation of sparse methods, we use two different methods to select keywords. First, we use the test and evaluation sets to mine keywords. We use TF-IDF to pull words and bigrams that are

most commonly found in train set articles about politics, but not found in off-topic articles. We take the top 40 words and bigrams and then sequentially pick those that maximise F1 on the evaluation set, until there is no remaining keyword that increases F1. Using this technique, the mined keywords were: vote, election, republican, committee, united, party, president, congress.

Second, we prompted Chat-GPT to produce keywords. We used the prompt: "You have a large number of 19th century US newspaper articles. You wish to classify these on whether they are about politics or not. The only way that you can do this is by checking whether they match any of a list of keywords or keyphrases. You can search for these keywords or phrases in each article, and if it matches any of them it will be classified as about politics, but if it does not match any, it will not. Please provide a list of keywords and phrases that will correctly classify as many of the articles that are about politics as possible, with a minimal number of off-topic articles classified as on topic" and received the following keywords: President, Congress, Election, Senator, Representative, Governor, Democratic Party, Whig Party, Republican Party, Suffrage, Legislation, Lawmakers, Government, Policy, Bill, Campaign, Debate, Vote, Political Convention, Public Office, Political Reform, Impeachment.

Using these lists of keywords, we consider any article to be predicted as on topic if it contains any of these keywords. We do not take case into account.

The structured data in `American Stories` allows us to classify at the article level, a significant advantage. However, for comparison with Chronicling America, we also evaluate the same methods at the scan level. A scan is counted as on topic if any article on that page is on topic. For neural methods on Chronicling America, we chunk the text into passages of 256 tokens, as the page OCR is significantly longer than the context window.

Content Dissemination Networks

To detect reproduced content, we also compare neural and sparse methods. In this case, the neural methods are only possible with `American Stories` , whereas sparse methods are possible with both `American Stories` and Chronicling America.

We evaluate these methods on all front pages from March 1, 1916, a randomly selected day. A single day is chosen because reproduced content tends to be published around the same time, so a single day will have a far higher number of reproduced articles than a random selection of front pages across time. On this day, `American Stories` contains 114 scans, with 2,354 articles. 1,994 of these were not reproduced, while 360 were reproduced. These 360 comprise 113 distinct articles, with the median reproduced article being reprinted 2 times.

For the neural method, we use the pre-trained neural model from [14]. This is a contrastively-trained bi-encoder finetuned from the MPNet Sentence BERT model [13, 16] on a large, hand-annotated sample of pairs of reproduced historical newspaper articles. [14] find this biencoder is marginally improved by running a cross-encoder over the outputs, but we do not reproduce these results as the cross-encoder is computationally costly for a small gain in performance. They also find that this fine-tuned biencoder model outperforms more generic semantic textual similarity models (eg. [13] by up to 20 percentage points. Thus the model is chosen to maximise accuracy, within a reasonable compute budget. At inference time, article representations are clustered using single

linkage clustering to detect reproduced content, and spurious links are pruned using community detection. We use the same distance threshold as in [14].

To enable a comparison to Chronicling America, where the content is only available at a page level, we amalgamate these results by page. A page-pair is counted as positive if they have any article in common. Nonetheless, the page-level evaluation of the neural method requires the data to be split into articles. It cannot be run over the unstructured text in Chronicling America.

Therefore for detecting reproduced content in Chronicling America, where we do not have article texts, we deploy the sparse methods from Viral Texts [15]. Viral Texts was designed specifically for detecting reproduced texts in Chronicling America's noisy page-level OCR by looking for overlapping n-gram spans. To compare this method between `American Stories` and Chronicling America, we also run it over the articles in `American Stories` and then amalgamate these results at the page level.

Finally, we deploy the locality-sensitive hashing (LSH) specification from [14] to evaluate the performance of sparse methods on `American Stories` , using the same parameters. In particular we do this because the Viral Texts method is not designed to be run at the article level. As expected, we find that LSH performs better than Viral Texts at the article level, but both methods perform significantly worse than the neural methods, in line with the findings of [14].

Story Clustering

Finally, we demonstrate that the content in `American Stories` can be clustered into news stories, following the same story between newspapers and across time. To create clusters of stories, we use a contrastively-trained biencoder.

We train this biencoder using data from `allsides.com`, a modern news website which shows how the same story is written by different newspapers. Articles on `allsides.com` are truncated. Groups of articles on the same story on `allsides.com` were used to create positive pairs. For negatives, we used the fact that each article is labelled with various tags, and also that we know which news source each article came from. For each article we take the article from the same news source, with different topic tags, that had the largest cosine similarity, using the biencoder before finetuning. In the cases where there were no articles with different tags from the same news source, we use articles from a news source which is lifted with the same political leaning. The specification that an article has different tags is important for making sure that articles are actual negatives.

Overall this gave 26,194 unique articles, with 18,382 positive pairs and 18,445 negative pairs. We featurized the data as "headline [sep] article" and we finetuned the biencoder from [14] as in experiments we found that this outperformed finetuning an MPNet Sentence BERT model [13, 16] directly. We optimised hyperparameters using hyperband [8]. The best model was trained fro 9 epochs, on a single GPU, with a batch size of 32, a warm up percent of 0.392. We optimised online contrastive loss [4], with and a loss margin of 0.497.

At inference time, we cluster using single-linkage clustering, with a cosine similarity threshold of 0.92. We control cluster size using leiden community detection [17]. We deduplicate the content using the method outlined in the section above. We take all articles that are reprinted at least five times, and run same story clustering over a year at a time.

Dataset details

Dataset URL

The dataset can be found at `https://huggingface.co/datasets/dell-research-harvard/AmericanStories`.

This dataset has structured metadata following `schema.org`, and is readily discoverable.[2]

Training labels for the individual models detailed in this paper are also available, and can be found at `https://huggingface.co/datasets/dell-research-harvard/AmericanStoriesTraining`.

DOI

The DOI for this dataset is: 10.57967/hf/0757.

License

The dataset has a Creative Commons CC-BY license.

Dataset usage

The dataset is hosted on Hugging Face. Each year in the dataset is divided into a distinct file. The dataset can be easily downloaded using the `datasets` library:

As the dataset is very large, files for specific years can be downloaded by specifying them or users can download all data for all years. Additionally, we provide two options for the output type. The first contains data at the article level, with features like newspaper name, page number, edition, date, headline, byline, and article text. The second contains data at the scan level. It contains information including the scan metadata; all detected content regions like articles, photographs, and adverts; legibility information, and bounding box coordinates.

```
from datasets import load_dataset

#  Download data for the year 1809 at the associated article level (Default)
dataset = load_dataset("dell-research-harvard/AmericanStories",
    "subset_years",
    year_list=["1809", "1810"]
)

# Download and process data for all years at the article level
dataset = load_dataset("dell-research-harvard/AmericanStories",
    "all_years"
)

# Download and process data for 1809 at the scan level
dataset = load_dataset("dell-research-harvard/AmericanStories",
    "subset_years_content_regions",
    year_list=["1809"]
)

# Download ad process data for all years at the scan level
dataset = load_dataset("dell-research-harvard/AmericanStories",
```

[2] See `https://search.google.com/test/rich-results/result?id=esZkoGgfOsLlnkrvwx9nSQ` for full metadata.

```
    "all_years_content_regions")
```

Users can find more information on accessing the dataset using the dataset card on Hugging Face.

Author statement

We bear all responsibility in case of violation of rights.

Maintenance Plan

We have chosen to host the dataset on huggingface as this ensures long-term access and preservation of the dataset.

Dataset documentation and intended uses

We follow the datasheets for datasets template [3]. Additionally, we have completed the dataset card on Hugging Face which can be accessed using the link to the dataset on Hugging Face hub [3]

Reproducibility

Moreover, we have included our responses to The Machine Learning Reproducibility Checklist [1] as outlined in table 2.

[3] https://huggingface.co/datasets/dell-research-harvard/AmericanStories

Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

The dataset was created to provide researchers with a large, high-quality corpus of structured and transcribed newspaper article texts from historical local American newspapers. These texts provide a massive repository of information about topics ranging from political polarization to the construction of national and cultural identities to the minutiae of the daily lives of people's ancestors. The dataset will be useful to a wide variety of researchers including historians, other social scientists, and NLP practitioners.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The dataset was created by a team of researchers at Harvard University, New York University, Northwestern Kellogg School, MIT, and Princeton University, led by Melissa Dell.

Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

Funding was provided by the Harvard Data Science Initiative, compute credits that Microsoft Azure provided to the Harvard Data Science Initiative, Harvard Catalyst, and the Harvard Economics Department Ken Griffin Fund for Research on Development Economics and Political Economy.

Any other comments?

None.

Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.

Dataset instances are detected content regions in newspaper page scans from the Library of Congress's Chronicling America collection. In the cases of article, headline, image caption, and byline regions, a text transcription is included if the page is written in English.

How many instances are there in total (of each type, if appropriate)?

Version 0.1.0 of American Stories contains 402 million content regions, 294 million of which include a text transcription.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

The Version 1.0 of the dataset will contain all possible instances. Version 0.1.0 contains approximately 40% of all instances as of 6/7/23.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.

Each instance includes: a unique content region id, its detected class (ARTICLE, HEADLINE, CAPTION, BYLINE, IMAGE, AD, TABLE, HEADER, PAGE NUMBER, or MASTHEAD), and the pixel coordinates of the newspaper page bounding box for the identified region. If the content region is classified as ARTICLE, HEADLINE, CAPTION, or BYLINE, the transcribed text is also provided.

Is there a label or target associated with each instance? If so, please provide a description.

Content regions are labeled by their model predicted class. Text article transcriptions have no label.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g. because it was unavailable). This does not include intentionally removed information but might include, e.g., redacted text.

No.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.

All articles and other content regions include metadata that can definitively determine relationships to other content regions. For example, two articles with the same lccn (newspaper identifier), edition, and page number are from the same newspaper page scan.

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

There are no recommended splits.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

Layout detection, OCR, and article association all introduce noise.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

The provided text data are self-contained. Some applications could require downloading the original scans, which are at `https://chroniclingamerica.loc.gov/`. All scans are freely available and in the public domain.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)? If so, please provide a description.

The dataset is drawn entirely from image scans in the public domain that are freely available for download from the Library of Congress's website.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.

Texts in the dataset reflect attitudes and values of a large, diverse group of newspaper editors and writers in the period they were written (1790-1960) and include content that may be considered offenseive for a variety of reasons.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

Yes. The dataset contains news about people.

Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.

It may be possible to infer certain characters about individuals covered in the news historically from the data. The authors of the dataset do not identify any subpopulations.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.

If an individual appeared in the news during this period, then texts may contain their name and other information. In some cases, it may be possible to link individuals to information on ancestry websites or Wikipedia (in the case of prominent historical figures). We do not attempt to do so in this paper.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description.

The data are drawn entirely from newspaper scans in the public domain.

Any other comments?

None.

Collection Process

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

The pipeline used to create layouts and article transcriptions from page images is described in detail within the paper. The dataset described here is the output of that pipeline.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?

The data extraction pipeline is described and evaluated in the main paper text.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

Release 1.0 will include everything in the Chronicling America scan collection.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

A large group of professors, research assistants, and students collaborated on all aspects of the data collection process, including labeling training data, training and validating models, data engineering, and conceptual design. All were compensated for their work, according to the regulations of Harvard University and New York University.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news

articles)? If not, please describe the timeframe in which the data associated with the instances was created.

Scans from Chronicling America were processed between 6/1/23 and 6/7/23. The data associated with the instances were created between 1780 and 1963, when they were published in local newspapers.

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

The data are entirely in the public domain and hence do not fall under the jurisdiction of university institutional review boards.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.

Yes, the articles in the dataset talk about people.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

We collected this data from a third party, the Library of Congress, which has verified that all data are in the public domain.

Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.

The data are in the public domain and cover many millions of individuals, most of whom are deceased.

Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.

The data are in the public domain and cover many millions of individuals, most of whom are deceased.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).

Not applicable.

Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.

No such analysis has been conducted.

Any other comments?

None.

Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of

instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.

No preprocessing was conducted.

Uses

Has the dataset been used for any tasks already? If so, please provide a description.

Example uses are detailed in the main text.

Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

No such repository currently exists.

What (other) tasks could the dataset be used for?

There are a large number of potential uses in the social sciences, digital humanities, and deep learning research, discussed in more detail in the main text.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?

This dataset contains unfiltered content composed by newspaper editors, columnists, and other sources. It reflects their biases and any factual errors that they made.

Are there tasks for which the dataset should not be used? If so, please provide a description.

We would urge caution in using the data to train generative language models - without additional filtering - as it contains content that many would consider toxic.

Any other comments?

None.

Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.

Yes. The dataset is available for public use.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub) Does the dataset have a digital object identifier (DOI)?

The dataset is available via HuggingFace. Download and instructions are at `https://huggingface.co/datasets/dell-research-harvard/AmericanStories`. The dataset's DOI is: https://doi.org/10.57967/hf/0757

When will the dataset be distributed?

The dataset is currently available.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

The dataset is distributed under a Creative Commons CC-BY license. The terms of this license can be viewed at `https://creativecommons.org/licenses/by/2.0/`

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.

There are no third party IP-based or other restrictions on the data.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.

No export controls or other regulatory restrictions apply to the dataset or to individual instances.

Any other comments?

None.

Maintenance

Who will be supporting/hosting/maintaining the dataset?

HuggingFace will continue to host the dataset. The authors will provide support, updates, and maintenance.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

Melissa Dell can be contacted via email at melissadell@fas.harvard.edu

Is there an erratum? If so, please provide a link or other access point.

There is no erratum.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?

The dataset will continue to be updated as new scans are processed. New versions will be added to HuggingFace. Anyone can subscribe to notifications about the dataset via HuggingFace.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.

All data are in the public domain.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users.

Older versions of the dataset will still be visable via the HuggingFace repo.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions

be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.

The dataset is privately created and maintained. There is no current plan to allow open source contributions.

Any other comments?

None.

Table 2: Reproducibility checklist

Item	Response	Comment
For all models and algorithms presented, check if you include:		
A clear description of the mathematical setting, algorithm, and/or model.	Yes	
A clear explanation of any assumptions.	Yes	
An analysis of the complexity (time, space, sample size) of any algorithm.	NA	Complexity can depend upon application-specific architecture and methods. We are using transformer-based models within our framework and we have reported the time profile and other related details
For any theoretical claim, check if you include:		
A clear statement of the claim.	Yes	
A complete proof of the claim.	Yes	
For all datasets used, check if you include:		
The relevant statistics, such as number of examples	Yes	
The details of train / validation / test splits	Yes	
An explanation of any data that were excluded, and all pre-processing step.	Yes	
A link to a downloadable version of the dataset or simulation environment	Yes	Link to Hugging Face Hub repo provided
For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control.	Yes	
For all shared code related to this work, check if you include:		
Specification of dependencies.	Yes	
Training code.	Yes	
Evaluation codes	Yes	
(Pre-)trained model(s).	Yes	Available on HuggingFace
README file includes table of results accompanied by precise command to run to produce those results	Yes	
For all reported experimental results, check if you include:		
The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results	Yes	
The exact number of training and evaluation runs.	Yes	
A clear definition of the specific measure or statistics used to report results.	Yes	

Continued on next page

Table 2: Reproducibility checklist (Continued)

A description of results with central tendency (e.g. mean) variation (e.g. error bars).	NA	
The average runtime for each result, or estimated energy cost.	Yes	Both training and inference times have been reported
A description of the computing infrastructure used	Yes	

References

[1] The Machine Learning Reproducibility Checklist (v2.0, Apr.7 2020).

[2] CARLSON, J., BRYAN, T., AND DELL, M. Efficient ocr for building a diverse digital history. *arXiv preprint arXiv:2304.02737* (2023).

[3] GEBRU, T., MORGENSTERN, J., VECCHIONE, B., VAUGHAN, J. W., WALLACH, H., AU2, H. D. I., AND CRAWFORD, K. Datasheets for datasets, 2021.

[4] HADSELL, R., CHOPRA, S., AND LECUN, Y. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (2006), vol. 2, IEEE, pp. 1735–1742.

[5] HOWARD, A., SANDLER, M., CHU, G., CHEN, L.-C., CHEN, B., TAN, M., WANG, W., ZHU, Y., PANG, R., VASUDEVAN, V., ET AL. Searching for mobilenetv3. *Proceedings of the IEEE/CVF international conference on computer vision* (2019), 1314–1324.

[6] JOHNSON, J., DOUZE, M., AND JÉGOU, H. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data 7*, 3 (2019), 535–547.

[7] KHOSLA, P., TETERWAK, P., WANG, C., SARNA, A., TIAN, Y., ISOLA, P., MASCHINOT, A., LIU, C., AND KRISHNAN, D. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362* (2020).

[8] LI, L., JAMIESON, K., DESALVO, G., ROSTAMIZADEH, A., AND TALWALKAR, A. Hyperband: A novel bandit-based approach to hyperparameter optimization, 2018.

[9] LIU, Y., OTT, M., GOYAL, N., DU, J., JOSHI, M., CHEN, D., LEVY, O., LEWIS, M., ZETTLEMOYER, L., AND STOYANOV, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[10] MUSGRAVE, K., BELONGIE, S., AND LIM, S.-N. Pytorch metric learning, 2020.

[11] OORD, A. V. D., LI, Y., AND VINYALS, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[12] PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., DESMAISON, A., KOPF, A., YANG, E., DEVITO, Z., RAISON, M., TEJANI, A., CHILAMKURTHY, S., STEINER, B., FANG, L., BAI, J., AND CHINTALA, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32* (2019), H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., pp. 8024–8035.

[13] REIMERS, N., AND GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[14] SILCOCK, E., D'AMICO-WONG, L., YANG, J., AND DELL, M. Noise-robust de-duplication at scale. *International Conference on Learning Representations* (2023).

[15] SMITH, D. A., CORDELL, R., AND MULLEN, A. Computational methods for uncovering reprinted texts in antebellum newspapers. *American Literary History 27*, 3 (2015), E1–E15.

[16] SONG, K., TAN, X., QIN, T., LU, J., AND LIU, T.-Y. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems 33* (2020), 16857–16867.

[17] TRAAG, V. A., WALTMAN, L., AND VAN ECK, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports 9*, 1 (Mar. 2019), 5233. Number: 1 Publisher: Nature Publishing Group.

[18] ULTALYTICS. Yolo v8 github repository. `https://github.com/ultralytics/ultralytics`, 2023.

[19] WIGHTMAN, R. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.