Performing Named Entity Recognition on Historical Text to Build Historical Timelines

CIS 9995 — Capstone Project Report

Nitin Vadnala | Advisor: Prof. Pei Wang | Temple University | Spring 2026

Abstract

The focus of this project was to build a Natural Language Processing framework that can perform Named Entity Recognition (NER) on historical text to automatically create chronological timelines. Two deep learning approaches are implemented and compared to one another: a fine-tuned DistilBERT transformer model and a Bidirectional LSTM with a Conditional Random Field (CRF) output layer. Both models are trained on news articles labeled via SpaCy and tested on five chapters from the U.S. History (OpenStax) textbook. The DistilBERT model was able to successfully identify all 79 unique years in the test data which matches the SpaCy pretrained baseline with a timeline-level F1 of 1.0. The BiLSTM-CRF model achieved a timeline F1 of 0.987 where it fell a little short in finding only 77 of the 79 unique years. Both models produced chronological sorted timelines which showcased the viability of NER-based automated timeline construction.

1. Introduction

Named Entity Recognition (NER) is a foundational concept in NLP where the goal is to identify and categorize specific entities pulled from unstructured text: people, organizations, places, dates, and other categories. This project applies NER to historical documents to automate the construction of chronological timelines where in traditional approaches would require large amounts of manual efforts by historians and researchers.

The motivation is based on how historians often need to manually read and annotate large amounts of text to identify key events and their temporal ordering. By automating entity extraction and chronological organization, this project showcases how modern NLP models can efficiently convert raw historical text into more structured, usable summaries. The system takes as input unstructured historical text and outputs a chronologically ordered timeline of events where each is assigned a year and the relevant sentence description.

Two different machine learning architectures are implemented and compared. The first is a DistilBERT-based transformer model, fine-tuned for token-level classification. The second is a Bidirectional LSTM with character-level CNN embeddings and a CRF output layer (BiLSTM-CRF). Both are evaluated against a SpaCy pretrained NER baseline using timeline-level precision, recall, and F1-score.

2. Related Work

NER has been massively studied in NLP for large periods of time. Early approaches were more based on rule-based systems and hand-crafted features. Statistical methods such as Hidden Markov Models and Conditional Random Fields (Lafferty et al., 2001) introduced structured prediction. The introduction of deep learning, mainly BiLSTM-CRF architectures (Huang et al., 2015), massively improved performance by learning feature representations automatically from data. Transformer-based pretrained language models like BERT (Devlin et al., 2019) and DistilBERT (Sanh et al., 2019) achieved top tier results on NER benchmarks such as CoNLL-2003.

Temporal tagging is a specific NER subtask focused on identifying time expressions. Notable systems include SUTime (Chang and Manning, 2012), a rule-based tagger, and neural approaches using CamemBERT and RoBERTa for temporal detection. However, most prior work focuses on identifying temporal expressions rather than using them to construct practical outputs like timelines. This project bridges that gap by using NER as a tool for automated timeline construction from historical text.

3. Methodology

3.1 Data Preparation

The training dataset consists of news articles passed through SpaCy's `en_core_web_sm` pretrained NER model to generate token-level BIO annotations. The BIO (Begin-Inside-Outside) tagging scheme marks the first token of an entity with a B- prefix and future tokens with I-, while non-entity tokens receive the O tag. This approach allows proper entity boundary detection which works much better than flat tagging where all tokens in an entity share the same label no matter the position.

The entity types included are: DATE, PERSON, ORG, GPE, NORP, LOC, EVENT, CARDINAL, ORDINAL, MONEY, TIME, WORK_OF_ART, LAW, QUANTITY, PERCENT, FAC, PRODUCT, and LANGUAGE. The complete tag set consists of 37 tags (18 entity types with B- and I- prefixes, plus O). The testing dataset is pulled from five chapters (chapters 21–25) of the U.S. History (OpenStax) textbook which covers about 145 pages of historical content with temporal references spanning from the 1600s to the 1900s.

3.2 DistilBERT Model

The first model uses a DistilBERT transformer encoder (Sanh et al., 2019) fine-tuned for token classification. DistilBERT is a distilled version of BERT that retains 97% of BERT's language understanding while being 60% faster and 40% smaller. The architecture adds a dropout layer ($p = 0.2$) and a linear classification on top of DistilBERT's 768-dimensional hidden representations.

An important detail to note is that subword token alignment should also be implemented here. DistilBERT uses WordPiece tokenization which may split a single word into multiple subword tokens. Only the first

subword of each original word gets the ground-truth BIO label while the upcoming subwords are assigned a special -100 index that is ignored by the CrossEntropyLoss function. During inference, predictions from the first subword are mapped back to the original word. Training uses AdamW optimization with a linear warmup schedule (10% of total steps), gradient clipping at norm 1.0, and batch size of 32 for 8 epochs.

3.3 BiLSTM-CRF Model

The second model is a Bidirectional LSTM with a Conditional Random Field output layer. The input representation joins word-level embeddings (dimension 100) with character-level CNN embeddings (dimension 50). The character CNN captures features useful for recognizing date patterns such as four-digit numbers. The final embeddings pass through a 2-layer BiLSTM (hidden dimension 256 per direction) with dropout ($p = 0.5$) aimed for the tag space by using a linear layer.

The CRF layer is the key architectural component. Rather than making independent tag predictions per token, the CRF models dependencies between adjacent tags via a learned transition matrix guaranteeing valid BIO constraints. For example, an I-DATE tag can only follow a B-DATE or another I-DATE. During training, the CRF computes the negative log-likelihood of the correct tag sequence. During inference, the Viterbi algorithm finds the globally optimal tag sequence. The model is trained for 20 epochs with batch size 64 and learning rate 0.001.

3.4 Timeline Generation

After NER inference, the timeline generation system pulls all token spans tagged as DATE, parses them to find all four-digit years, links each year with its source sentence, deduplicates entries, and sorts them chronologically. The date parser handles multiple formats: bare years (1848), month-year combinations (March 1848), full dates (July 4, 1776), and decade references (the 1920s).

4. Experiments and Results

4.1 Training

The training dataset was split 90/10 into training and validation subsets.

Figure 1 shows the DistilBERT loss curves. Training loss drops from 1.08 to about 0.13 within the first two epochs and then continues to decrease to around 0.02 by epoch 8. Validation loss levels out near 0.11 after epoch 2 and remains flat indicating the model reaches its best generalization early. The slight gap between training and validation loss in later epochs points to a bit of overfitting though validation loss does not increase, so the model is still strong.

Figure 2 shows the BiLSTM-CRF loss curves. Both training and validation loss decrease steadily over all 20 epochs, starting near 23 and 16 respectively and converging toward negative values (which is expected

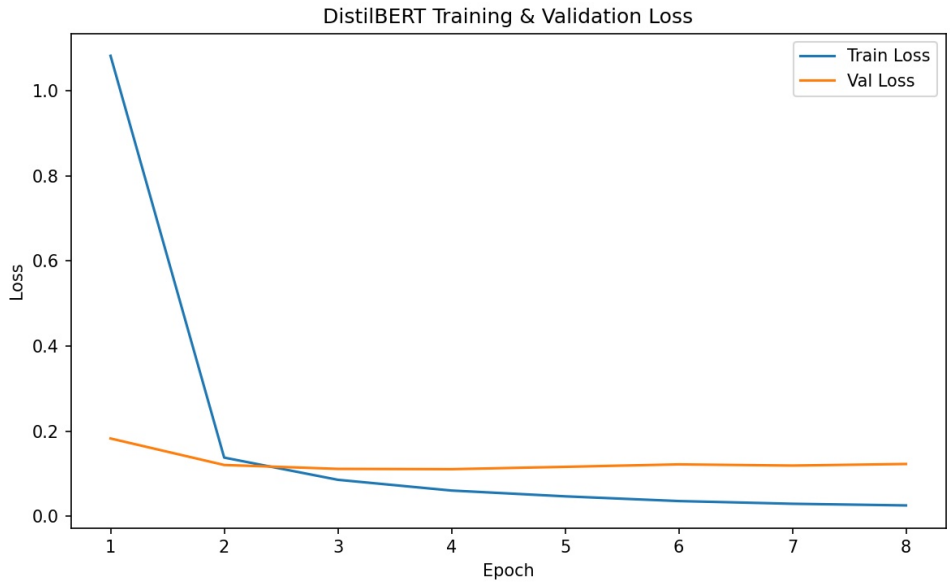

Figure 1: DistilBERT training and validation loss over 8 epochs.

for CRF log-likelihood loss, where higher log-probabilities produce negative loss values). The training and validation curves track each other closely with a small consistent gap indicating good generalization without overfitting.

4.2 Timeline Evaluation

Table 1 shows the timeline-level evaluation results. The test chapters contain 79 unique years across 420 timeline-relevant sentences. All three approaches are evaluated on how many of those years they correctly extract and whether the outputted timelines are chronologically sorted.

Table 1: Timeline-level evaluation. "Entries" = total timeline sentences extracted. "Years" = unique years found vs. reference. Precision, recall, and F1 are computed at the unique-year level.

Model	Entries	Years	Prec.	Recall	F1	Sorted
SpaCy	420	79/79	1.000	1.000	1.000	Yes
DistilBERT	473	79/79	1.000	1.000	1.000	Yes
BiLSTM-CRF	439	77/79	1.000	0.975	0.987	Yes

4.3 Analysis

The DistilBERT model matched the SpaCy baseline perfectly in year coverage by identifying all 79 unique years in the test data. It actually produced more timeline entries (473 vs. 420) meaning it found more contextual sentences with the extracted dates. This means that a fine-tuned DistilBERT model can match

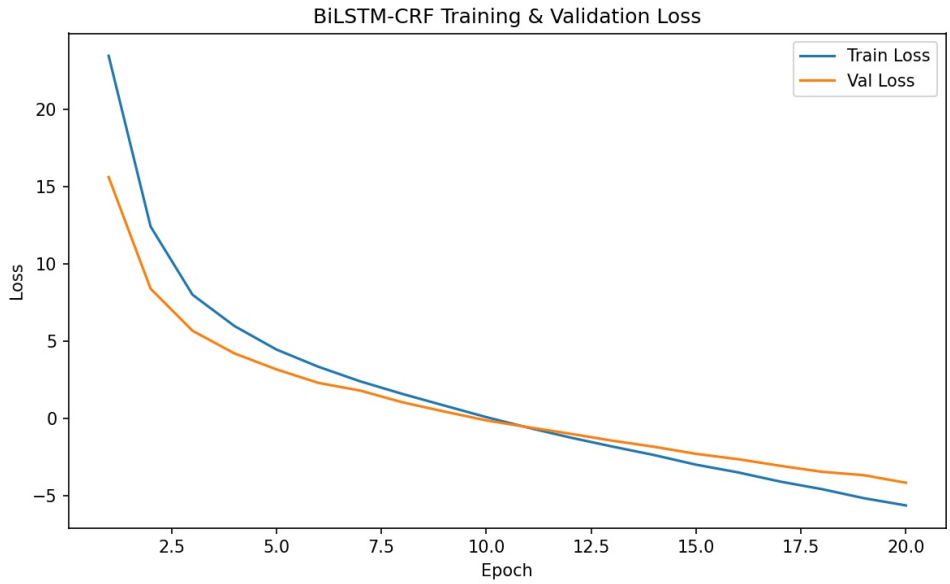

Figure 2: BiLSTM-CRF training and validation loss over 20 epochs.

or exceed a pretrained SpaCy pipeline on temporal entity extraction when the training data is sufficiently representative.

The BiLSTM-CRF model captured 77 of 79 unique years (recall = 0.975), missing two years from the test data. Precision being perfect at 1.0, meaning every year it did extract was valid and the model did not hallucinate any false dates. The two missed years are edge cases involving unusual date formatting or contexts that the model's character-level and word-level embeddings could not capture. Even with the slightly lower recall, the BiLSTM-CRF still produced a strong timeline with 439 entries with all correctly sorted chronologically.

All three approaches produced perfectly sorted timelines confirming that the chronological ordering step of the pipeline is achievable. The fact that both trained models achieve perfect precision across all extracted years means that the BIO tagging scheme and entity extraction pipeline reliably find temporal entities from other tokens.

5. Key Improvements Over Prior Work

The most significant architectural improvement is the addition of the CRF layer to the BiLSTM model. Without a CRF, the original LSTM made independent predictions per token which would commonly produce wrong tag sequences. The CRF's learned transition matrix eliminates these kinds of errors. Character-level CNN embeddings give the BiLSTM-CRF model the ability to recognize date patterns based on character morphology rather than relying just on word-level semantics.

On the DistilBERT side, proper subword label alignment ensures that the loss function is only done on meaningful tokens. The warmup scheduler stabilizes early training by gradually increasing the learning rate

Table 2: Summary of improvements over the prior implementation.

Aspect	Previous Version	Current Version
Tagging	Flat tags (PERSON, DATE, O)	BIO scheme (B-DATE, I-DATE, O)
DistilBERT	Basic linear head only	Dropout + warmup scheduler + sub-word alignment
LSTM Model	2 training examples, no CRF	Full BiLSTM-CRF with char-CNN embeddings
Evaluation	Manual comparison of ~10 dates	Automated metrics on 420+ entries
Date Parsing	Only YYYY format	Multiple formats, decade references
Code	Notebooks with hardcoded paths	Modular Python package with config

which is seen in the smooth convergence visible in the loss curve. The evaluation pipeline is able to use automated metrics rather than manual spot-checking.

6. Conclusion

This project demonstrates that NER-based timeline construction is a viable and effective approach to automating temporal information extraction from historical text. The DistilBERT model matched the SpaCy pretrained baseline by identifying all 79 unique years in the test data and producing a richer timeline with more contextual entries. The BiLSTM-CRF model achieved a timeline F1 of 0.987 demonstrating that even without pretrained transformer representations, a well-designed sequential model with structured prediction can perform well.

Future work could explore: (1) expanding the date parser to handle relative temporal expressions such as “a decade later” or “at the turn of the century”; (2) training on domain-specific historical corpora rather than news articles to reduce potential domain mismatch; (3) incorporating a temporal normalization step to resolve ambiguous dates; and (4) building an interactive web interface that allows users to upload historical text and receive a visual timeline.

References

Chang, A. X. and Manning, C. D. (2012). SUTime: A Library for Recognizing and Normalizing Time Expressions. *LREC 2012*.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL 2019*.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv:1508.01991*.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *ICML 2001*.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108*.

Stroetgen, J. and Gertz, M. (2012). Temporal Tagging on Different Domains: Challenges, Strategies, and Gold Standards. *LREC 2012*.