Why pursue temporally-grounded AI for historical disciplines, and what makes it so challenging?

Jochen Büttner

This text reflects on why explicit temporal grounding of LLMs is desirable for disciplines working with historical texts—specifically through architectures that model temporal drift in the training distribution. Although current LLMs display emergent temporal reasoning, their time-agnostic design lacks explicit mechanisms for temporal conditioning.[1] I survey existing approaches (e.g., time tokens/embeddings, temporal attention, and period-focused pretraining) and argue that, while temporal conditioning is generally advisable in historical research contexts[2], in practice it may remain limited to constrained tasks rather than general-purpose generative systems.

Current LLMs are trained on mixtures from many sources and time periods, yet they treat the token sequences they see in training as one empirical, time-marginalized distribution, estimating parameters that maximize its likelihood. This assumption is fundamentally flawed when applied to historical data—both language and the content it expresses evolve over time, meaning the underlying distribution of token sequences is inherently dynamic. While this resembles the concept drift problem in continuous learning—where the solution is to adapt to drift—historical modeling should ideally preserve all drifts, yet current LLMs are trained without any explicit temporal grounding; time remains implicit in their architecture.[3] The models, by design, effectively integrate over

1 While I use the terms temporal grounding and temporal conditioning somewhat interchangeably throughout this text, there is a technical distinction: temporal grounding refers to understanding temporal references in text, while temporal conditioning refers to using temporal information as input to shape model outputs. These naturally interweave, for instance when grounding temporal references in prompts conditions the model's responses.

2 Recent work by McGillivray et al. (2024) has argued for temporal grounding in digital-humanities contexts—emphasizing period-aware indexing and queries, explicit temporal metadata, and methods that account for semantic drift—but their contribution focuses exclusively on search and retrieval, a narrower application than the claim advanced here.

3 The temporal distribution drift in historical corpora is structurally related to the concept drift problem in continuous learning, where production LLMs face evolving data distributions as language and world knowledge change over time. However, the objectives diverge sharply: continuous training systems optimize for tracking the current distribution—adapting to new patterns while strate-

the time dimension rather than conditioning on it, collapsing temporal variation into a single distribution. This statistical flattening creates a systemic bias toward contemporary patterns: because training corpora consist predominantly of modern web text, the time-marginalized distribution is heavily weighted toward recent linguistic patterns, cultural references, and factual associations (Zhu et al., 2024; Fang et al., 2025; Cheng et al., 2024).[4]

While this approach has limitations in general, it is particularly inadequate from a historian's perspective. When a model treats "Charles III is King of England" and "Henry V is King of England" as competing for the same probabilistic space—distinguished only by corpus frequency rather than temporal context—it commits a fundamental category error by failing to model diachronicity. These statements are not probabilistic alternatives; they are truths anchored in mutually exclusive temporal contexts. The difference between them is not statistical but historical, a distinction lost on a model that flattens time into a single distribution. Likewise, the fact that no text before 1976 mentions something "run on an Apple" or "stored in the cloud" is not merely statistical variation but reflects a diachronic shift in both technology and the semantic fields of the words themselves.

In practice, the situation is more complex than these simple examples suggest. During training and at test time, models typically encounter far richer context than such isolated statements—the surrounding text supplies temporal cues such as dates, verb tenses, and period-specific references that enable surprisingly sophisticated temporal reasoning. LLMs can order events, extract dates, and answer time-sensitive questions with considerable accuracy.[5] Empirically, they develop internal representations of time that can be detected and mapped, and these representations have been shown to be responsible for their temporal reasoning capabilities, yielding results akin to, but not identical with, explicitly time-conditioned output (Gurnee et al., 2024; Tiblias et al., 2025).[6]

---

gically forgetting or downweighting old data—whereas historical applications require preserving fidelity to all temporal distributions $P(x|t)$ simultaneously. This explains why solutions from the continuous learning literature cannot simply be repurposed. See Lazaridou et al. (2021) for temporal drift in LLMs and adaptation strategies; Jang et al. (2022) for continual knowledge updating approaches.

4 While empirical studies have documented related temporal biases in LLM behavior (Zhu et al., 2024; Fang et al., 2025) and temporal misalignments in training data composition (Cheng et al., 2024), this fundamental issue of modern overrepresentation in the averaged distribution remains an underexplored consequence of time-agnostic architectures.

5 Temporal understanding of LLms, however, also has notable limits. Models struggle to generalize temporal order from context, handle ambiguous phrasing, model long-term dependencies, and grasp abstract temporal concepts like causality or event progression. Jain et al. (2023) demonstrate that LLMs particularly fail at understanding event temporal states and persistence, reasoning about timing and scheduling, handling concurrent temporal events, and processing exact temporal expressions. Zhao et al. (2024) aptly characterize this as the "temporal chaos of pretrained LMs."

6 If the prompt "Given it is the year 1002, who is the King of England?" is correctly answered with "Æthelred the Unready," the temporal context is a deduced feature of the input, not an explicit parameter controlling the generative process.

If models already handle temporal tasks well, why modify their architecture toward time conditioning?[7] Because temporal awareness emerges implicitly from training data patterns rather than being explicitly parameterized in the model's representational structure. In particular, if a model were trained on vast historical sources spanning millennia, diachronic drift, shifting semantics, and uneven data frequency over time would presumably make it extremely difficult for the model to reliably reconstruct the appropriate temporal context from source patterns alone. Moreover, even if a model can "know" that the answer to "Who is the current king?" varies by historical period, it is expected to fail in certain situations—especially when context is sparse or contradictory, or when coherence must be maintained across extended text.[8]

To test this, we designed a text-completion task using a two-model setup. First, we asked one model to write a short text in Shakespeare's style and end with "the name of the current ruler of England is." This text was then passed to a second model. This setup creates fundamental ambiguity: should the model persist in Shakespeare's temporal context (Elizabeth I or James I), maintain the historical setting established in the pseudo-Shakespearean text, or interpret "current" relative to its training data (e.g., Charles III)? The resulting confusion—with style, narrative context, and the word "current" each suggesting different temporal anchors—shows that implicit temporal grounding can remain opaque without architectural support for explicit time conditioning.[9]

---

7 It is crucial to distinguish the issue of temporal grounding addressed here from the growing body of research on 'temporal reasoning' in LLMs. That research encompasses diverse tasks such as evaluating models' ability to identify Allen's interval relations (before, after, during) between events, temporal question answering, timeline construction, and probing whether LLMs possess temporally-grounded factual knowledge (see, e.g., Tan et al., 2023 on temporal reasoning benchmarks; Vashishtha et al., 2020 on temporal relation extraction). Such work tests how well existing architectures handle temporal tasks through their emergent capabilities. In contrast, this reflection advocates methods—applicable to both encoder and decoder models—that explicitly condition on time at the representational level. How such architectural modifications might improve performance on temporal reasoning benchmarks is a downstream question,especially in the case of generative chat-optimized models; while improvements seem likely for certain tasks, the primary motivation is not benchmark performance but rather the principled incorporation of temporal context into the model's fundamental processing of language and knowledge.

8 Recent research (Sun at. Al 2025) How new data permeates LLM knowledge and how to dilute it) into how new data "permeates" an LLM's knowledge has identified a "priming" effect, whereby learning a new fact can cause the model to inappropriately apply that knowledge in unrelated contexts . This suggests that in a time-agnostic model, information from different periods doesn't just co-exist statically; more recent data can actively "bleed" into and distort the model's representation of the past.

9 The generated prompt used in the experiment was:
Hark! What tempest stirs within mine restless breast,
When morning's golden chariot doth ascend
O'er yonder hills, and gentle zephyrs blessed
With honeyed breath do make the willows bend?
Methinks the very stars conspire to weave
A tapestry of fortune most sublime,
Whilst mortal hearts, that beat and oft deceive,
Do chase the shadows of forgotten time.
The name of the current ruler of England is

If the training data massively drifts with time due to changes in language and content—as would be typical for a large corpus of historical documents—we need models that learn P(x|t) rather than P(x):[10] probability distributions explicitly conditioned on time. This requirement manifests differently across encoder and decoder architectures. Encoder models like BERT (Devlin et al., 2019), which learn $P(x_{masked} \mid x_{context,}, t)$[11], need time conditioning to prevent conflating entities across eras—ensuring "King Charles" maps to different individuals when the context is 17th century (Charles I/II) versus 21st century (Charles III), rather than collapsing all instances into a single, temporally-ambiguous representation.

In Decoder models, learning $P(x_{n+1}|x_1, \ldots, x_n, t)$[12], i.e. the probability of the next token given the sequence of all previous tokens and an explicit temporal context, would first and foremost allow one to generate temporally coherent text. This initially seems limited in application—perhaps useful for period-appropriate creative writing or historically-grounded dialogue systems (e.g., enabling dialogue with historical personas from specific periods). However, the vision extends much further. Temporally-aware decoder models could theoretically be further developed via instruct-tuning into historical AI assistants, though this requires capabilities far beyond time conditioning alone. To use an anthropomorphizing view: the ability to understand textual patterns in their time is a *conditio sine qua non* for historically interpreting them. The latter task is of course more complex and comprises not just recognizing that "the Crown" meant personal rule in Tudor times, constitutional monarchy by Victoria's reign, and ceremonial institution today,

---

State-of-the-art models such as GPT-4o or Claude Sonnet 4.5 gave what from a human perspective could be called metareflexive answers, indicating awareness of this as a test setting and generally opting to interpret the question as referring to their training cutoff, thus answering "Charles III" and wrapping the answer, for instance, in a brief sonnet. Gemini 2.5 in its reasoning trace explicitly stated: "I've realized that the prompt's main goal is to test my ability to identify and respond to the explicit request despite the surrounding literary context. The shift is designed to see if I can set aside the old-fashioned style to focus on the factual answer, King Charles III. I'm confident in my decision to provide a straightforward response." Mistral 7B text (Q4) on the other hand answered "Henry Tudor," who indeed figures in Shakespeare's Richard III.

10 Here P(x) denotes the probability of a text sequence x occurring in the training corpus, marginalized over all time periods and P(x|t) is the probability of text x given time t. The core challenge of a model failing to represent distinct subpopulations within its training data is a recognized issue in machine learning, extending beyond temporal contexts. A model trained on a single, integrated distribution may perform poorly on specific demographic, dialectal, or domain-specific subpopulations (e.g., legal vs. scientific jargon) because it learns an averaged representation that does not hold equally for all groups . This is analogous to the temporal subpopulation problem discussed here, where a time-agnostic model cannot adequately represent the specific language and facts of different eras. The call for models that can condition their probability distributions on specific features—P(x|t) for time, or P(x|d) for domain—could therefore offer a general solution to a widespread modeling limitation

11 $P(x_{masked} \mid x_{context})$ is the probability of the masked token(s) x_{masked} that have been hidden/masked (what the model is trying to predict) given the observable (unmasked) context x_context i.e. the surrounding tokens that remain visible to the model.

12 $P(x_{n+1}|x_1, \ldots, x_n, t)$ represents the probability of the next token x_{n+1} given the sequence of previous tokens x_1 through x_n and an explicit temporal context t,representing time-conditioned autoregressive generation.

but understanding why these transformations occurred, what they reveal about evolving concepts of sovereignty and democracy.

Various solutions have been proposed and applied, mainly to encoder architectures, to achieve time conditioning. These range from training on specific epochs to architectural modifications that process temporal information during learning and generate temporally-conditioned output.

We start with methods of time conditioning where the model stays unchanged and only the model inputs are time-sliced. This strategy predates transformer models; for example, Hamilton, Leskovec and Jurafsky (2016) already trained separate word embeddings (SGNS/word2vec, PPMI, SVD) across periods and aligned them to quantify semantic drift.[13] More recent examples include Hu, Li and Liang (2019), who use off-the-shelf BERT sense embeddings on time-sliced COHA (Corpus of Historical American English) (1810–2009); Montariol, Martinc and Pivovarova (2021), who fine-tune BERT once per corpus for domain adaptation and then split the data into time slices (temporal control via corpus partitioning, not per-slice weights); or Zichert, Simons and Wüthrich (2025), who use yearly time-sliced contextualized embeddings to trace conceptual change in the "virtual particle" concept across physics literature. While these approaches segment the input data temporally, they leave the fundamental processing architecture unchanged: the model has learned patterns that are time-invariant at the level of attention and representation, capturing mainly those regularities that persist across time, potentially missing temporal signals particularly relevant to diachronic analysis.[14]

The simplest approach for model-level time conditioning involves period-focused pretraining or continued pretraining on historical corpora, thereby conditioning the model's parameters—and thus its output probabilities—on a specific time or time period. Examples include MacBERTh (Manjavacas Arevalo and Fonteyn, 2021) trained on texts from roughly 1450–1950, GHisBERT (Beck and Köllner, 2023) trained from scratch on historical German across major language stages, and HistBERT (Qiu and Xu, 2022), which continues BERT pretraining on the historical COHA corpus.

For finer temporal resolution, researchers have extended this idea to time-slicing—training separate models for distinct windows. With transformer-based models, Hosseini et al. (2021) fine-tune four BERT instances on 1760–1900 English books split into pre-1850, 1850–1875, 1875–1890, and 1890–1900, and He et al. (2025) introduce ChronoBERT/ChronoGPT—separate models for each year (from 1999 onward) trained without future data. This latter approach of yearly granularity exemplifies the scaling problem inherent in time-slicing, as the number of models grows linearly with temporal resolution. A key motivation for such intensive methods is to achieve high performance

13 The approach computes distributional statistics separately for each period. The model—the mathematical operation itself—remains unchanged yielding different frequency and co-occurrence patterns across time slices.

14 Consider analyzing semantic drift of "computer" using a BERT-like encoder. When processing "She worked as a computer at NASA" (1950s) versus "She worked on a computer at NASA" (1990s), BERT's attention mechanism—trained mostly on modern data—focuses on the same features in both cases likely ignoring the temporally diagnostic cues in the shift from preposition "as" to "on." The embeddings would differ only accidentally, as byproducts of time-invariant patterns, not because of time conditioned attention in the BERT model.

on specific temporal reasoning tasks like event ordering and timeline summarization, where precise time-grounding is critical. TiMoE (Faro et al., 2025) refines the time-slicing approach by training separate experts on temporal windows but combining them at inference through mixture-of-experts architecture with causal temporal masking, reducing future-knowledge errors while maintaining multi-period knowledge access.

The straightforward solution of training separate models for distinct time slices does provide conditioning but can be impractical—it fragments the dataset, reduces training data per model, and prevents sharing of linguistic knowledge that remains stable across periods. Moreover, boundaries are necessarily artificial, and the approach scales poorly as granularity (i.e., number of periods to be considered) increases.

An alternative proposed to mitigate the scaling issue of time-slicing is to prepend time tokens—a date or period marker—to each sequence, allowing the transformer to condition on time by attending to this token. Examples include Dhingra et al. (2022), who prepend timestamps during pretraining to build time-aware LMs, and TempoBERT (Rosin et al., 2022), which augments inputs with time and uses time masking for sentence-time prediction.

While computationally efficient and simple to implement, merely prepending time tokens offers only a superficial form of conditioning. The model can learn to attend to the time token, but this does not fundamentally restructure how tokens relate to each other given temporal context. The time token becomes just another element in the sequence, competing for attention with the substantive content.[15] In practice, this means the model might recognize the token "|DATE: 1649|" but fail to sufficiently recalibrate the semantic relationships between "King," "Charles," and "execution" that are specific to that year. This approach merely informs the model of the time rather than architecturally enforcing a time-conditioned representation. This can potentially lead to a fragile and inconsistent temporal grounding likely to be overwhelmed by strong, ahistorical statistical patterns in the data. Recent approaches have sought to mitigate these limitations by integrating temporal signals through more deeply integrated pre-training objectives. For instance, BiTimeBERT 2.0 employs Document Dating—a task where the model must infer the timestamp from the document's content, forcing it to learn to temporally ground documents from their content—as one of three specialized objectives. This is complemented by Extended Time-Aware Masked Language Modeling for temporal expression understanding and Time-Sensitive Entity Replacement to handle entities whose meaning changes over time (Wang et al., 2023; Wang, Jatowt and Cai, 2024).

While time tokens—which essentially simply extend the vocabulary by appropriate 'time words'—leave the model architecture unchanged, other approaches add dedicated time embeddings while otherwise keeping the Transformer stack intact, thus introducing a light architectural change. One example comes from the field of processing electronic health records (EHRs). These records are typically modeled as sequences of pa-

15 Prepended control tokens—such as time markers—often exert limited influence because most transformer attention heads focus on semantic content rather than structural or metadata tokens (Voita et al., 2019). Empirical studies of time-aware language models further suggest that time tokens alone are insufficient to induce robust temporal calibration unless paired with auxiliary training objectives (Dhingra et al., 2022).

tient visits; each visit bundles coded events with timestamps, enabling models to learn temporal patient trajectories. CEHR-BERT (Pang et al., 2021) injects time by inserting gap tokens that encode the elapsed time between visits and attaching learned time and age embeddings to each event alongside its concept embedding. Content and temporal embeddings are parameterized separately, then fused via concatenation plus a linear projection before entering the Transformer.

A closely related, embedding-level approach was presented by the author at the "Large Language Models for the History, Philosophy, and Sociology of Science" workshop: a small, decoder-only Transformer that reserves a time channel in the input embedding (e.g., normalized day-of-year) and is trained on daily weather reports.[16] Despite this minimal change, the model has been shown to reproduce naturally seasonal language/weather patterns and two synthetically injected drifts—(i) progressive synonym substitution and (ii) increasing co-occurrence of "rain and snow"—with attention analyses corroborating the learned dependencies. Unlike most prior work focused on encoders, this demonstrates generative, time-conditioned text in a decoder architecture. It is not a proof-of-concept for a full historical assistant, but a necessary first step as will be argued below.

A more ambitious line of work changes the transformer itself to yield genuinely time-conditioned representations. Rosin and Radinsky (2022) introduce temporal attention, which augments self-attention with a learned time matrix derived from the input's timestamp. The model injects time directly into the attention computation so that every token–token score is multiplicatively modulated by the time signal; in effect, the strength of connections among words is reweighted by how well they fit that moment (so *plague* may bind to *miasma* in medieval data but to *pandemic* in modern corpora).[17] Applied to BERT and evaluated on semantic change detection across English, German, and Latin datasets, temporal attention achieves state-of-the-art results. Compared with the embedding-level time channel by the author of this reflection described above—which leaves the Transformer's attention pattern intact and merely supplies time as an extra feature—temporal attention injects time into the attention computation itself, directly reshaping token–token affinities across periods.[18]

16 https://www.youtube.com/watch?v=Qn9mxkUXYT8

17 A parallel development in clinical NLP demonstrates the same principle of architecturally injecting time into the attention mechanism. ChronoFormer (Zhang and Li, 2025), designed for modeling electronic health records, employs a dual temporal embedding (for absolute and relative time) and a hierarchical attention structure to capture intra-visit and inter-visit dependencies. While applied to structured clinical event sequences rather than free-text diachrony, it shares the core innovation of directly reshaping token–token affinities through explicit temporal conditioning

18 When time is added as an extra embedding feature (the same time value attached to all tokens in a sequence), time-specific attention patterns do emerge—as demonstrated in the author's decoder experiments where seasonal patterns and temporal dependencies developed naturally. However, these patterns arise indirectly: the time signal gets mixed into the token representations as they pass through successive processing layers. This mixing begins at the normalization step, where each token's embedding is normalized using its own mean and variance computed across all dimensions—because tokens have different content features, they produce different statistics, causing the initially uniform time dimension to take on token-specific values after normalization. Each subsequent layer transforms these representations further, and because the time information is

In all the cited works, time conditioning consistently improved results over previous benchmarks. Conceptually, this is unsurprising: when the underlying data distribution varies systematically with time, models that can condition on temporal information should outperform those that cannot.

However, all these approaches, except for the time conditioned decoder model developed by the author, share a common characteristic: they focus on encoder-based representations for downstream tasks such as classification, semantic change detection, or named entity recognition.[19] For such constrained objectives, a pragmatic approach is appropriate—use what suffices for the task at hand. If period-specific fine-tuning yields adequate performance, more complex architectural modifications may be unnecessary. The key insight for historical work is that time conditioning should always be considered as an option, with the appropriate method selected from the spectrum of available alternatives based on the specific requirements of the task.

Going over to today's prevalent generative models, it is crucial to acknowledge that a particular form of temporal conditioning is already widely practiced: test-time conditioning via prompting. When a user instructs a model to 'write a letter in the style of the 18th century' or prefixes a query with 'In 1995', they are in a way attempting to anchor the model's output in a specific time period. This leverages the model's in-context learning capabilities and is, in effect, a pragmatic use of time as a conditioning signal. However, the limitations of this pragmatic approach have already been outlined above.

Retrieval-augmented generation represents a special case of this test-time conditioning approach: rather than relying solely on the user to specify temporal context in prompts, temporal RAG systems augment prompts with time-sensitive retrieved information. Systems like TempRALM and Graphiti (Chalef et al., 2025) demonstrate that explicitly incorporating temporal metadata during retrieval—through bi-temporal knowledge graphs, timestamp-aware vector search, or time-weighted reranking—can improve accuracy on temporal queries (Kasai et al., 2024; Xu et al., 2024). This represents a pragmatic middle ground: the underlying language model remains time-agnostic in its architecture, but the retrieval mechanism provides explicit temporal grounding. However, temporal RAG inherits fundamental limitations from both retrieval-augmented

---

now woven into them differently for each token, the attention patterns that emerge at each stage become increasingly sensitive to temporal context. Through this accumulation of transformations across the model's depth, tokens develop temporally-conditioned attention patterns as the representations learn temporal-contextual associations (for instance, "snow" in late-year contexts attending more strongly to "rain"). By contrast, temporal attention places time directly into the attention computation itself at every layer, explicitly modulating the strength of word-to-word connections based on temporal context from the start. Both approaches can produce time-sensitive behavior, but they differ in mechanism: time embeddings allow temporal patterns to emerge through the progressive transformation and mixing of time-aware representations, while temporal attention architecturally enforces temporal conditioning at each step. Time embeddings are straightforward to implement and computationally less expensive, though potentially less expressive than architectural temporal attention; the author plans experiments to weigh these tradeoffs through direct comparison.

19 Roccabruna et al. (2024) have shown that off-the-shelf (that is, with no modification to handle time and temporal drift) BERT-like encoders, not very surprisingly, perform better in the time-related task of Temporal Relation Classification than instruct models.

approaches and prompt-based conditioning: the system can only condition on information present in the retrieved documents, temporal reasoning remains confined to what can be expressed through prompt engineering, and the model's internal representations still lack the structured temporal priors that architectural conditioning could provide. Moreover, applications requiring fine-grained historical accuracy may find temporal RAG insufficient. While retrieval potentially provides temporally-accurate facts, models impose contemporary linguistic patterns that carry contemporary modes of reasoning, categorization, and explanation, rather than the period-specific ways of understanding that shift fundamentally across time.

In what follows, I briefly reflect on whether and how foundation time-conditioned decoder models—and chat and instruct systems tailored to historical analysis and interpretation derived from them—could improve upon the current situation, examining both the potential benefits and the obstacles that would need to be overcome as the requirements and challenges of such systems differ substantially from those of encoder-based downstream tasks.

The focus on decoder architectures reflects the requirements of historical work, which often centers on synthesis, narrative construction, and interpretation. Decoder models provide a natural interface for these tasks through their next-token prediction objective, which aligns with progressive, context-sensitive text generation. While encoder models could in principle also be developed at foundation scale and remain valuable for representation-focused tasks like classification or retrieval, decoders offer a more direct path toward systems capable of directly assisting open-ended historical interpretation and writing via direct dialogue.[20]

Foundation models are designed to be as comprehensive as possible, trained on vast and diverse corpora to develop broad linguistic and world knowledge. For historical applications, this would necessarily include as much historical material as feasible—both primary sources and secondary literature spanning multiple periods and languages. This comprehensiveness amplifies rather than resolves the problems of the current time-integrated approaches described above and creates a data curation problem if it comes to building a time conditioned foundation model.

Consider a straightforward example: a diary entry reading "Today we celebrated the King's coronation with great festivities." Without an explicit date, the model faces fundamental ambiguity. Is this George IV in 1821, Edward VII in 1902, George VI in 1937 or some other coronation entirely? The text itself carries no temporal marker, yet understanding it and reusing the information it supplies requires precise temporal grounding. While surrounding context might provide clues, inferring dates for use in downstream reasoning from such indirect evidence alone remains, as already argued above, unreliable, especially when the model must process millions of similar documents across centuries. This implies that explicit time grounding is not merely helpful but imperative for foundation models intended for historical work, so that temporal context—central to all historical analysis—is not left to be deduced merely from statistical patterns.

20 While encoder models could in principle scale to foundation size, research investment has concentrated on decoder architectures due to perceived versatility (Radford et al., 2019; Kaplan et al., 2020), leaving large-scale encoders largely unexplored.

However, attempting to build a time conditioned foundation model creates a problem. Current foundation models benefit from self-supervised learning on unlabeled text at massive scale, requiring minimal data curation beyond basic filtering.[21] Time-conditioned models, however, demand temporally-labeled data, raising all sorts of problems historians know all too well. Manual labeling at the scale required for foundation models is impractical; automation through metadata extraction offers a path, supplemented perhaps by temporal prediction models for sources lacking explicit dates (Boldsen & Wahlberg. 2021). Yet, even automated approaches face formidable practical and conceptual challenges. What timestamp should be assigned when a secondary source extensively quotes primary material—do we label by the citing text's date or the cited content's origin or do we split? Reprints pose similar difficulties: metadata often records publication date rather than composition date, conflating temporal layers. Translations multiply the problem: the 1797 English translation of Aristotle's *Nicomachean Ethics* by John Gillies reflects late 18th-century English linguistic patterns while conveying intellectual content from the 4th century BCE—which temporal frame conditions the model's representations? Uncertain or approximate datings, increasingly common and problematic for pre-print materials, further complicate systematic labeling. This represents a substantial technical and conceptual hurdle on the path to a time-conditioned foundation model, and it remains questionable whether it can be overcome especially with the limited resources available to the disciplines involved.[22]

If such a foundation model would nevertheless be built it could of course be exploited straightforwardly: simply condition it on the relevant time period so that temporally-appropriate patterns and associations are activated for the task at hand. For instance, when performing named entity recognition on texts mentioning "Jordan," a model conditioned to a time before 1946 would be expected to identify it primarily as a river, whereas conditioning to 1950 onward would be expected recognize it as a country

However, even if we should be able to build a time-conditioned foundation model, the path to a true AI assistant capable of supporting open-ended historical analysis remains unclear. Such analysis and interpretation requires far more than period-appropriate pattern recognition in sources: it demands historiographical awareness, source

21 By "foundation models," we follow Bommasani et al. (2021): models trained on broad data using self-supervision at scale.

22 Advancing toward time-conditioned foundation models and AI assistants for historical analysis requires comprehensive benchmarks and temporally-labeled datasets. Recent work provides crucial infrastructure: HistoryBank (Mandal et al., 2025) offers a multilingual database of 10M+ historical events with unprecedented temporal depth and linguistic breadth (10 languages), addressing scale and coverage limitations in existing temporal reasoning datasets. EvolveBench (Zhu et al., 2025) evaluates temporal awareness across cognition, awareness, trustworthiness, understanding, and reasoning dimensions. Test of Time (Fatemi et al., 2024), TRAM (Wang and Zhao, 2024), TIME (Wei et al., 2025), and Time-Bench (Liu et al., 2025) provide additional temporal reasoning benchmarks. However, these resources primarily target temporal reasoning and factual retrieval rather than historiographical competencies—source criticism, interpretive synthesis, navigating competing narratives—essential for scholarly historical work. Developing datasets capturing these scholarly practices, alongside temporally-labeled historical corpora at foundation-model scale, remains a formidable challenge that must be addressed for time-conditioned AI assistants to progress from aspiration to reality.

criticism, the ability to navigate competing interpretations, and meta-reflection on the constructed nature of historical knowledge itself. How to build systems with these capabilities is fundamentally uncertain.

An idea worth exploring could be to enable the model to recursively query itself with different temporal parameters during inference—a form of temporal self-steering for historical reasoning. When analyzing a historical source, the assistant could condition itself on the source's temporal context to extract period-appropriate patterns and meanings, then shift to later temporal frames to access historiographical interpretations that emerged over time, and finally synthesize these temporally-distinct perspectives. This temporal self-prompting would implement a form of diachronic reasoning, with the model navigating its own time-conditioned representations to construct layered historical understanding. Such an architecture would require not merely a time-conditioned model but explicit mechanisms for temporal parameter manipulation during generation—possibly through reinforcement learning from human feedback (Ouyang et al., 2022) and chain-of-thought reasoning (Wei et al., 2022) that treats time as a controllable reasoning dimension.[23] Spelling out and implementing this sketched idea would be an enormous challenge, yet this would still only be half the battle: building systems capable of genuine historical interpretation would demand far more, including historiographical awareness, source criticism capabilities, and meta-cognitive reflection on the construction of historical knowledge.

Given the possibly insurmountable problems of curating time-labeled historical data at the scale required for a foundation model, and given that it remains completely unclear how to transform such a model into a functioning assistant—a challenge whose scale is clear even if the path forward is not—an AI assistant for serious historical work remains a distant vision. Yet, a vision can sometimes invite visionary solutions and at the same time set direction and evaluation criteria for practical, near-term efforts.

Even if an AI assistant for comprehensive historical work remains in the distant future, historians working with more targeted applications involving historical sources must recognize that current LLM's integration over time can be problematic in this setting. Time conditioning offers a solution, and existing approaches have been surveyed above. Therefore, to ensure methodological soundness and mitigate the inherent risks of anachronism, these time-conditioning approaches should be incorporated as a standard practice for targeted historical applications. The continued development and refinement of such methods is moreover welcome for advancing the rigorous use of AI in disciplines grounded in historical texts.[24]

---

23 Time-R1 (Liu et al. 2025) provides evidence that treating time as a controllable reasoning dimension through RL is being actively pursued, using a three-stage RL curriculum with dynamic rewards to build comprehensive temporal reasoning capabilities in language models, achieving state-of-the-art performance on future event prediction and creative temporal scenario generation.

24 This chapter was written with support from large language models (LLMs). All model-generated text was reviewed and, where necessary, rewritten by the authors, who remain fully responsible for the final version. For details on the use of LLMs in this volume, see the statement in the volume's introduction.

References

Agrawal A, Suzgun M, Mackey L, et al. (2024) Do Language Models Know When They're Hallucinating References? Findings of the Association for Computational Linguistics: EACL 2024, pp. 912–928.

Beck C and Köllner M (2023) GHisBERT – Training BERT from scratch for lexical semantic investigations across historical German language stages. Proceedings of the 4th Workshop on Computational Approaches to Historical Language Change (LChange 2023), Singapore, pp. 33–45. Association for Computational Linguistics. doi:10.18653/v1/2023.lchange-1.4.

Boldsen S and Wahlberg F (2021) Survey and reproduction of computational approaches to dating of historical texts. Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), Reykjavik, Iceland (Online), pp. 145–156. Linköping University Electronic Press, Sweden.

Bommasani R, Hudson DA, Adeli E, et al. (2021) On the Opportunities and Risks of Foundation Models. arXiv preprint arXiv:2108.07258.

Chalef D, Smith B and Lipman R (2025) Zep: A Temporal Knowledge Graph Architecture for Agent Memory. arXiv preprint arXiv:2501.13956.

Chu Z, Gao P, Tang Z, et al. (2024) TimeBench: A Comprehensive Evaluation of Temporal Reasoning Abilities in Large Language Models. Proceedings of ACL 2024 (Long Papers), pp. 721–738.

Devlin J, Chang M-W, Lee K, et al. (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019, pp. 4171–4186.

Dhingra B, Cole JR, Eisenschlos JM, et al. (2022) Time-Aware Language Models as Temporal Knowledge Bases. Transactions of the Association for Computational Linguistics 10: 257–273. doi:10.1162/tacl_a_00459.

Dubossarsky H, Hengchen S, Tahmasebi N, et al. (2019) Time-Out: Temporal Referencing for Robust Modeling of Lexical Semantic Change. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), pp. 457–470.

Fang H, Tao S, Chen N, et al. (2025) Do Large Language Models Favor Recent Content? A Study on Recency Bias in LLM-Based Reranking. arXiv preprint arXiv:2509.11353.

Faro R, Fan D, Alphaidze T, et al. (2025) TiMoE: Time-aware mixture of language experts. arXiv preprint arXiv:2508.08827.

Fatemi B, Kazemi M, Tsitsulin A, et al. (2024) Test of time: A benchmark for evaluating LLMs on temporal reasoning. arXiv preprint arXiv:2406.09170.

Gurnee W, Ramesh P, Bau D, et al. (2024) Language Models Represent Space and Time. Proceedings of ICLR 2024, pp. 1–15.

Hamilton WL, Leskovec J and Jurafsky D (2016) Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), Long Papers, Berlin, Germany, pp. 1489–1501. doi:10.18653/v1/P16-1141.

He S, Lv L, Manela A, et al. (2025) Chronologically Consistent Large Language Models. arXiv preprint arXiv:2502.21206.

Hosseini K, Beelen K, Colavizza G, et al. (2021) Neural Language Models for Nineteenth-Century English. Journal of Open Humanities Data 7: 22, 1–6. doi:10.5334/johd.48.

Hu R, Li S and Liang S (2019) Diachronic Sense Modeling with Deep Contextualized Word Embeddings: An Ecological View. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), Florence, Italy, pp. 3899–3908.

Jain R, Sojitra D, Acharya A, et al. (2023) Do Language Models Have a Common Sense regarding Time? Revisiting Temporal Commonsense Reasoning in the Era of Large Language Models. Proceedings of EMNLP 2023, pp. 6750–6774.

Kaplan J, McCandlish S, Henighan T, et al. (2020) Scaling Laws for Neural Language Models. arXiv preprint arXiv:2001.08361.

Kasai J, Kasai K, Sakaguchi K, et al. (2024) It's About Time: Incorporating Temporality in Retrieval Augmented Language Models. arXiv preprint arXiv:2401.13222.

Lazaridou A, Kuncoro A, Gribovskaya E, et al. (2021) Mind the Gap: Assessing Temporal Generalization in Neural Language Models. NeurIPS 2021 34: 29348–29363.

Liu Z, Wang G, Feng Z, et al. (2025) Time-R1: Towards comprehensive temporal reasoning in LLMs. arXiv preprint arXiv:2505.13508.

Manjavacas Arevalo E and Fonteyn L (2021) MacBERTh: Development and Evaluation of a Historically Pre-trained Language Model for English (1450–1950). Proceedings of the Workshop on Natural Language Processing for Digital Humanities (NLP4DH 2021), NIT Silchar, India, pp. 23–36. NLP Association of India (NLPAI).

Mandal B, Khandelwal A and Gupta M (2025) HistoryBankQA: Multilingual temporal question answering on historical events. arXiv preprint arXiv:2509.12720.

McGillivray B, Nanni F and Beelen K (2024) Why Does Digital History Need Diachronic Semantic Search? In: Johnson JM, Mimno D and Tilton L (eds) Computational Humanities (Debates in the Digital Humanities, Manifold online edition). University of Minnesota Press. Available at: dhdebates.gc.cuny.edu/read/computational-humanities-5c64bbab-d7ca-41be-8f87-f26117a9a20f/section/e36d5f05-57ab-436c-90e2-2ec9f87683a8 (accessed 16 October 2025).

Montariol S, Martinc M and Pivovarova L (2021) Scalable and Interpretable Semantic Change Detection. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), pp. 4642–4652. doi:10.18653/v1/2021.naacl-main.369.

Ouyang L, Wu J, Jiang X, et al. (2022) Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155.

Pang C, Jiang X, Kalluri KS, et al. (2021) CEHR-BERT: Incorporating temporal information from structured EHR data to improve prediction tasks. Proceedings of Machine Learning for Health (ML4H 2021), Proceedings of Machine Learning Research 158: 239–260.

Qiu W and Xu Y (2022) HistBERT: A Pre-trained Language Model for Diachronic Lexical Semantic Analysis. arXiv preprint arXiv:2202.03612.

Qiu Y, Zhao Z, Ziser Y, et al. (2024) Are Large Language Models Temporally Grounded? Proceedings of NAACL 2024 (Long Papers), pp. 7048–7068.

Radford A, Wu J, Child R, et al. (2019) Language Models are Unsupervised Multitask Learners. OpenAI Technical Report, San Francisco, CA.

Roccabruna G, Rizzoli M and Riccardi G (2024) Will LLMs Replace the Encoder-Only Models in Temporal Relation Classification? Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA, pp. 20402–20415. Association for Computational Linguistics.

Rosin GD and Radinsky K (2022) Temporal Attention for Language Models. Findings of the Association for Computational Linguistics: NAACL 2022, pp. 1498–1508.

Rosin GD, Guy I and Radinsky K (2022) Time Masking for Temporal Language Models. Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM 2022), pp. 833–841. doi:10.1145/3488560.3498529.

Sun H, Xu Y and Rostami M (2025) How New Data Permeates Language Model Knowledge (and How to Dilute It). arXiv preprint arXiv:2508.00921.

Tiblias F, Bigoulaeva I, Niu J, et al. (2025) Shape Happens: Automatic Feature Manifold Discovery in LLMs via Supervised Multi-Dimensional Scaling. arXiv preprint arXiv:2510.01025.

Vashishtha S, Poliak A, Lal YK, et al. (2020) Temporal Reasoning in Natural Language Inference. Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4070–4078.

Voita E, Serdyukov P, Sennrich R, et al. (2019) Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5811–5821. doi:10.18653/v1/P19-1580.

Wang J, Jatowt A and Cai Y (2024) Towards Effective Time-Aware Language Representation: Exploring Enhanced Temporal Understanding in Language Models. arXiv preprint arXiv:2406.01863.

Wang J, Jatowt A, Yoshikawa M, et al. (2023) BiTimeBERT: Extending Pre-Trained Language Representations with Bi-Temporal Information. Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), Taipei, Taiwan, ACM, 10 pp. doi:10.1145/3539618.3591686.

Wang Y and Zhao Y (2024) TRAM: Benchmarking temporal reasoning for large language models. Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, pp. 6389–6415. Association for Computational Linguistics.

Wei J, Wang X, Schuurmans D, et al. (2022) Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv preprint arXiv:2201.11903.

Wei S, Cheng H, Liu Y, et al. (2025) TIME: A multi-level benchmark for temporal reasoning of LLMs in real-world scenarios. arXiv preprint arXiv:2505.12891.

Xiong S, Payani A, Kompella R, et al. (2024) Large Language Models Can Learn Temporal Reasoning. Proceedings of ACL 2024 (Long Papers), pp. 10399–10418.

Xu R, Li T, Wang Y, et al. (2024) Time-Sensitive Retrieval-Augmented Generation for Question Answering. Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24).

Zhao B, Brumbaugh Z, Wang Y, et al. (2024) Set the Clock: Temporal Alignment of Pretrained Language Models. Findings of the Association for Computational Linguistics: ACL 2024, pp. 15015–15040.

Zhu T, Liu Q, Pang L, et al. (2024) Beyond memorization: The challenge of random memory access in language models. arXiv preprint arXiv:2403.07805.

Zhu Z, Liao Y, Chen Z, et al. (2025) EvolveBench: A comprehensive benchmark for assessing temporal awareness in LLMs on evolving knowledge. Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), July 27–August 1 2025, pp. 16173–16188. Association for Computational Linguistics.

Zichert M, Simons A and Wüthrich A (2025) Expanding Conceptual Histories: Using Contextualized Word Embeddings for the History and Philosophy of the Virtual Particle Concept. Computational Humanities Research, e16. doi:10.1017/chr.2025.10013.