Retrieving Time from Scanned Books

John Foley and James Allan

Center for Intelligent Information Retrieval
University of Massachusetts Amherst
Amherst, MA

Abstract. While millions of scanned books have become available in recent years, this vast collection of data remains under-utilized. Book search is often limited to summaries or metadata, and connecting information to primary sources can be a challenge.

Even though digital books provide rich historical information on all subjects, leveraging this data is difficult. To explore how we can access this historical information, we study the problem of identifying relevant times for a given query. That is – given a user query or a description of an event, we attempt to use historical sources to locate that event in time.

We use state-of-the-art NLP tools to identify and extract mentions of times present in our corpus, and then propose a number of models for organizing this historical information.

Since no truth data is readily available for our task, we automatically derive dated event descriptions from Wikipedia, leveraging the both the wisdom of the crowd and the wisdom of experts. Using 15,000 events from between the years 1000 and 1925 as queries, we evaluate our approach on a collection of 50,000 books from the Internet Archive. We discuss the tradeoffs between context, retrieval performance, and efficiency.

1 Introduction

With the growing number of digital libraries and the growing size of digital collections, a vast number of historical documents have been made publicly available. For example, the Internet Archive has over six million books available for free online[1]. These books are available in many languages, are from many cultures and time periods, and cover all subjects. Information retrieval in this broad historical domain is an important and interesting challenge. We believe that time is the key to success on many interesting tasks in this domain, and the first step to better use the historical information in these documents is to extract and predict times.

An example of a query with time *explicitly* specified is `lincoln april 14 1865` whereas a query for `lincoln assassination` would have that same temporal connotation, but with time *implicitly* included. In this work, we will refer to these kinds of information needs as "events" where the event here is the assassination of President Abraham Lincoln in 1865. There are many other details about

[1] `https://archive.org`

this event, such as the name of theater, the assassin, and all of these details are part of this event which is often described in text. Given a query describing all or part of an event, we hope to retrieve a specific piece of information assigned to it: the date or set of dates on which it occurred.

In this work, we consider events to be the basis for information needs. Query log analysis from related work suggests that events, or at least temporally motivated queries are common in web search: Metzler et al. [19] report that queries with implicit temporal facets are close to 7% of web queries. Nunes et al. automatically detect temporal expressions in 1.5% of web queries and also report low recall [20].

In digital books in particular, preliminary findings suggest that temporal information is critical. The Hathi-Trust is an academic resource that allows full-text searching within books. A random sample of 600 queries from their logs shows that about 10% of these queries contain a date or time facet [26]. This suggests that dates are more important for book search than for general web search, and that the percentage of queries for which there are unspecified times could be quite high.

In addition to looking at events with a single, obvious time point, we are interested in the case where a user only has a partially-specified information need, or they are interested in a class of events. Consider a user interested in the early history of Deerfield, Massachusetts, who might enter a query like `raid on deerfield`. The implicit time association of this query is ambiguous, whether the user is aware of that or not, and may refer to either of the following events:

February 29, 1704
French and Native American forces attacked the English frontier settlement at Deerfield, Massachusetts during Queen Anne's War.

September 12, 1675
Deerfield was sacked during King Phillip's War, and "the people as had not been butchered fled to Hatfield" [24, p. 272].

An ideal search system would be able to present the user with information to help understand and reformulate their search results with respect to time. Such a system would allow a user interested in the raids of Deerfield to consider either or both relevant dates in the collection through query reformulation.

In section 3, we discuss how we use state-of-the-art NLP tools to extract temporal information from our corpora. In section 4, we propose several models for organizing this extracted data as retrievable events, and unsupervised methods for predicting years from the highest scoring events. In section 5, we look at how to automatically derive a gold standard for this task. We discuss the results of our evaluation on 50,000 digitally-scanned books (16 million pages) in section 6. We show that our new, hybrid model is the most effective while resulting in about 14% smaller retrieval indices.

2 Related Work

This study is motivated in part by other work that addresses the task of selecting a time for queries when none is explicitly specified. Metzler et al. [19] identify the problem of recognizing queries that have implicit year qualification—for example, the name of a conference where there is often a user need for the particular year. They resolve the year ambiguity using an approach that mines query logs. We explore this issue for collections where this no query log available as well as where the range of potential years is substantially greater.

Campos et al. [4] look at date-tagging web queries by collecting the set of words that co-occur in snippets gathered from a commercial search engine containing candidate dates and choosing dates based on the most similar set.

Kanhabua and Nørvåg [13] address this challenge by using "time language models" directly with pseudo-relevance feedback (PRF), and indirectly using the publication dates of the PRF documents. Results from their intrinsic evaluation demonstrate that using publication dates significantly outperforms the language-model baseline. In contrast, we find that publication dates are not helpful in our larger and "messy" corpus.

2.1 Finding Times for Documents

Another line of research that is very similar to the task we study here is estimating dates for *documents*, typically aiming to identify the correct publication date. Kanhabua and Nørvåg [12] improve on de Jong et al.'s document date-finding techniques [7] by interpolating between adjacent time quantiles, using entropy to give higher weight to discriminative terms, and additionally smooth their language models using day-by-day query statistics available from Google Zeitgeist.

Kumar et al. build language models of years based on Wikipedia biography pages in order to estimate the focus time of other documents [15].

Jatowt et al. [10] use co-occurrence based statistics to determine words that are strongly associated by times and use the words that make up documents to determine what times are related to documents. They used a large corpus of news documents to date a small corpus of Wikipedia, book, and web timeline events.

2.2 Other Uses of Time in IR

The value of time as a dimension for general information retrieval is well studied [1, 2]. As one example, there is a lot of work that tries to leverage time expressions or time in order to improve retrieval [3, 6, 11]. These papers highlight the importance of having correct dates associated with a query or document, motivating our work to understand the best techniques for *finding* those dates when they are missing or suspect.

Much work on time expressions can be traced back to TimeML, an XML-based format for expressing explicit, implicit, and relative references to time in text. [22]. In this study, we focus on explicit references to time, as resolving the relative references is often based on publication date, and implicit references are generally culturally-specific (e.g. the Christmas Season).

There is little work on retrieving times in archival or historical documents. Smith considers the task of detecting and browsing events, using documents with times manually annotated by historians [23]. In contrast, we explore approaches to a different task using only automatically annotated times.

Language modeling is a standard and state of the art approach to general information retrieval tasks [21], as well as those in the time domain [16]. The basis of our approach is language modeling and the sequential dependence model [18] which incorporates term dependencies from adjacent query terms.

Question answering (QA) is an area of Information Retrieval that works toward constructing natural language responses for natural language questions. Often, the simplest technique applied to QA tasks is a form of passage or sentence retrieval [25], although much modern work is focused on creating and exploiting structured resources [8]. Our evaluation task will turn out to be very similar to QA restricted to "when" questions, though is differently motivated and we consider a broader spectrum of approaches than is used by the QA community.

3 Extracting Temporal Information

We ran the Stanford CoreNLP toolkit [17], version 3.3.1, on our book corpus, yielding sentence boundaries and date/time expressions (along with other annotations that we did not use for this study).

Since we were not investigating the task of extracting times, our processing decisions focused on precision over recall. To this end, we ignored all relative time expressions ("last year", "next Christmas") rather than introduce normalization errors. For the same reason, we kept only sentences that contained exactly one (absolute) time expression, minimizing the ambiguity by avoiding any sentences referring to multiple events.

While we wanted to use fine-grained time information, it was rare in this corpus, so we focused on years alone. While 71% of time expressions extracted had years, less than 20% of those had a day or a month included.

As an example of how the time expressions relate to the topics, consider Sylvester's work describing one of the Deerfield Massacres [24]. He mentions the year 1704 in 23 sentences, and Deerfield is mentioned in only five of them. The entire corpus mentions that year in 10,176 sentences. In all sentences with an absolute time reference, "Deerfield" is mentioned just 643 times.

4 Methods

4.1 Event Modeling for Retrieval

In this work we propose a number of ways to model events using our time-tagged documents. We describe these models, the intuition behind each, and how they were evaluated against an input query event.

Sentence-Event Model The SENTENCE-EVENT model we propose uses the hypothesis that every sentence mentioning a time describes a unique event. To evaluate this model, we treat each sentence as a separate document and use state-of-the-art baseline retrieval methods to rank them in relation to our queries.

Document-Event Models Another model to consider is one that assumes every *book* discusses a single event. This is our BOOK-EVENT model. Certainly, there are many books that fit this model, i.e. ones discussing a civil war battle in depth, but there many history books that cover numerous such events. As a result of this, we also considered a model that assumes every *page* of every book would describe an event, called the PAGE-EVENT model.

These models are similar to those used in systems like these that run on newswire collections. In such collections, the assumption that an article discusses a single event is more intuitively correct: since such publications are often much shorter and more focused. In related work, these models were much stronger than their counterparts in the books collection we used here.

To evaluate the BOOK-EVENT model and the PAGE-EVENT model, we treat each book or page as an independent document, and rank them.

Year-Event Models Since our task involves predicting the year of an event as a query, it makes sense to try and model all the events within a single year directly, and simply use this aggregate model to predict the best year for each input query.

This approach was used by many others, as there has been substantial work proposing using time-based language models as a means of retrieving times or time expressions [7, 12, 13, 15].

To evaluate our YEAR-EVENT models, we construct a language model from all the sentences mentioning a particular year, and rank the years by their similarity to the language model of our query events.

Book-Year-Event Models The model we propose in this work unifies the intuition between the BOOK-EVENT and the YEAR-EVENT models. Since books are likely to be topically coherent, the assumption that all events corresponding to the same year will be similar is more likely to be valid within the context of a *single* book than across all books. This has the advantage of being between the too-few events of the YEAR-EVENT approach and the too-many events of the BOOK-EVENT, PAGE-EVENT, or especially SENTENCE-EVENT approach.

To evaluate this approach, we grouped our sentences containing unique, absolute time references into models by originating document and year pairs. These models were then ranked by similarity to the query events.

4.2 Year Ranking and Prediction

Regardless of the event-modeling framework, we need to rank our event models (and thus possible years) by the input event query. As we mentioned before, we use two popular, state-of-the-art baseline retrieval methods: query likelihood (QL) [21] and the sequential dependence model (SDM) [18].

QL is a unigram approach, like those that have been studied in the literature for retrieving relevant times [4, 7, 12, 13]. The markov-random-field model of term dependencies in SDM is consistently a top performer in standard retrieval metrics across collections, so we present results using both techniques.

In evaluation of all of the models except the Year-Event models, we have the possibility for multiple event models to predict the same year. For example, with the Year-Book-Event models, we might retrieve two books discussing the sinking of the Titanic, with reference to the same year.

Although our initial model considers these as separate events, we can use the multiple-hypotheses generated by the highest-scoring models in order to improve our prediction.

To ensure applicability to the long-tail of history, we treated this problem of selecting years from high-scoring event models as an unsupervised re-ranking problem. We only discuss the most successful method here, briefly, for space reasons. *Reciprocal Rank Weighting* was used, which assigns every occurrence of a year a score equal to 1/rank, and sums them across all occurrences. That means that a year that occurs at ranks 1, 3 and 4 would achieve a score of $1 + 1/3 + 1/4$. This approach is based on the intuition that multiple occurrences are important, but less important as you travel down the ranked list.

4.3 Evaluation Metrics

We evaluate queries with a single relevant year with mean reciprocal rank (MRR). Since there is only one relevant year, it makes sense to use this metric as it directly measures the rank of the relevant document (year). MRR is a common evaluation for question answering [25], and fits this class of queries well.

We evaluate queries with multiple relevant years with normalized discounted cumulative gain, or NDCG [9]. NDCG allows for graded relevance judgments, but in our experiments we only considered binary relevance: 1 if a year was related to that query, and 0 otherwise. The results we show are the mean NDCG across all queries. Evaluating with average precision (AP) gave us similar results, so we do not include it here for simplicity and for space reasons.

5 Collecting Queries

This work is based on the idea that it is valuable to know the time—year or years, specifically—that are related to a query. In a typical retrieval task, a system's job is to rank documents in an order that reflects the chance that they are relevant to the information need. In contrast, in this study our task is, given a query, to rank *years* by the chance that they are relevant to the information need.

To understand which approaches are most effective at finding the correct years, we need queries and corresponding years. The question answering corpora from past community evaluations [25, 5] include a small number of questions that have year as an answer. Unfortunately, there are only a handful of such questions and even fewer that overlap with the time periods of our document sets. (Most focus on modern news or Web corpora.)

To create a large number of queries we turn to Wikipedia. Nearly every year has a "year page" within Wikipedia that lists events, births, and deaths within that year, typically with references to Wikipedia pages with additional details. For example, as of Summer 2014, the year page for 1704 (`http://en.wikipedia.org/wiki/1704`) lists 13 events with a specific month or date (e.g., "September: War of the Spanish Succession" and "February 29: Raid on Deerfield (Queen Anne's War): French-Canadians and Native Americans sack Deerfield, Massachusetts, killing over 50 English colonists."), 9 events with unknown date within the year (e.g., "Isaac Newton publishes his *Opticks*"), 14 births, and 20 deaths.

For our purposes, we only consider the "Events" sections of the Wikipedia year pages. This means that we discard all dates of birth or death. We also ignored any year pages 2014 and higher, which are the future of our June 2013 english XML dump.[2].

5.1 Queries with a Single Relevant Year

We converted all Wikipedia markup into plain text and extracted all event entries. We removed any facts that were made up entirely of stop words[3] and we explicitly removed from the entry any numbers that could be year or day references. We also removed the mention of months for those entries that had them. Our goal in that processing was to remove all mentions of dates other than the entry's corresponding year, which was used only as the relevance judgment for that entry as a query.

That processing resulted in 40,356 facts with associated years, spanning 560 B.C. through A.D. 2013. Table 1 shows some example events. For the one-year task, where the goal is to select the correct single year for a query, we used the event description directly without further processing.

Based on the domain of years extracted from our test collection, we down-sampled these years to only those that were actually discussed in our books: roughly 1000-1925 A.D.

5.2 Queries with Multiple Relevant Years

Some queries are ambiguous with respect to time, an issue we aim to also explore in this work. In order to have queries that were relevant to a *set* of years, we merged similar queries from different years as follows.

As mentioned previously, the events in the Wikipedia year pages contain links to articles the discuss the entities involved in an event. If two events from different years link to exactly the same pages, then then there is temporal

[2] `http://dumps.wikimedia.org/enwiki`
[3] We used the Lemur 418 stopword list from `http://lemurproject.org/`

Table 1. Example queries with a single relevant year.

Year	Fact
1178	The Sung Document is written, detailing the discovery of "Mu-Lan-Pi" (suggested by some to be California) by Muslim sailors.
1298	Residents of Riga and the Grand Duchy of Lithuania defeat the Livonian Order in the Battle of Turaida
1535	Manco Inca Yupanqui, nominally Sapa Inca, is imprisoned by the Spanish Conquistadors of Peru.
1704	French-Canadians and Native Americans sack Deerfield, Massachusetts.
1733	British colonist James Oglethorpe founds Savannah, Georgia.

ambiguity regarding that collection of pages. We therefore grouped all one-year queries by co-occurring links, creating one query from many. For example, consider the following three one-year queries from Wikipedia year pages, with links to Wikipedia articles shown in small caps:

- (1221) The MAYA of the YUCATÁN revolt against the rulers of CHICHEN ITZA.
- (1528) The MAYA peoples drive SPANISH CONQUISTADORES out of YUCATÁN.
- (1848) The Independent Republic of YUCATÁN joins Mexico in exchange for Mexican help in suppressing a revolt by the indigenous MAYA population.

Those three entries all mention "Maya" and "Yucatán" so we join them together. To generate a query, we selected just their common words — in this case, the query *maya, yucatán*. Table 2 shows several other resulting queries and their multiple relevant years. As the random sample in this table shows, the majority of these generated queries had only two relevant years, although a few had more.

Table 2. Example queries with a multiple relevant years.

Years	Shared Terms
1221, 1528, 1848	yucatán, maya
1862, 1863	battle, general, ambrose, confederate, civil, war, american, union, burnside
1700, 1721	pope, xi, succeeds, innocent, clement
1380, 1382	horde, tokhtamysh, blue, khan, golden, mamai
1916, 1821	republic, colombia, venezuela
1588, 1577	spanish, plymouth, francis, drake

6 Results & Discussion

This set of experiments was run on a collection of 50,228 scanned books taken from the the Internet Archive,[4]. Rather than select books at random, we chose to use the books selected by the INEX book track [14], to simplify reproducibility. In order to generalize to millions of other books available online, we did not use any of the structured XML information provided for those challenges.

Some of the books in the collection are quite large. On average, there are 86,871 terms in a book, and about 270 terms on each page. There are 16 million pages in this collection.

Of the 14.9 million time expressions that were extracted by the tagger, 10.2 million contain an absolute year reference. Queries were generated as discussed in Section 5 and were distributed to a 1/3 training, 2/3 evaluation split. There were 15,739 single year queries distributed evenly by year. There were a total of 3,235 queries with multiple relevant years and they were distributed randomly.

6.1 Results

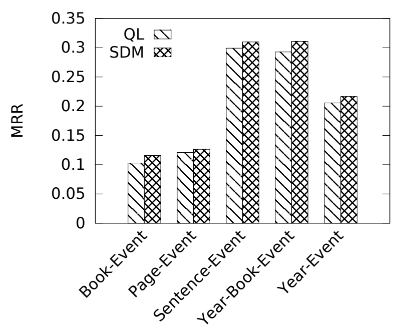

Fig. 1. MRR on queries with 1 relevant year.

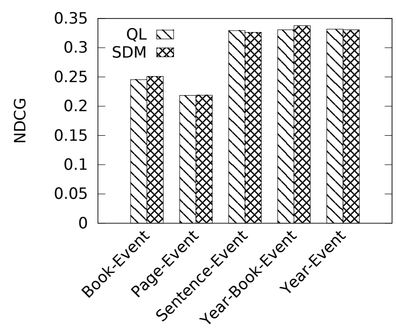

Fig. 2. NDCG on queries with two or more relevant years.

Our results for single-year queries on the books corpus are shown in Figure 1. We find that the Year-Book-Event and the Sentence-Event models were most effective. Of notice is that the Year-Event model performed poorly in comparison on queries with a single relevant year, suggesting that events occurring in the same year suffer from being being lumped together, such that all events are included and rare events may be overshadowed by the countless mentions of popular events that year.

While document neighbors were found to be effective in related work, both Book-Event and Page-Event performed poorly in comparison to the other models. The poor performance of Book-Event was to be expected, as there

[4] `https://archive.org`

are 188 years mentioned per book on average, and so it makes sense that the intuition of one event per book is incorrect, however, the poor performance of PAGE-EVENT is more surprising, as there was only a single date on every other page on average.

We note that when there is a single mention of a year on a page, the year is represented by its containing sentence in SENTENCE-EVENT but by all text on the page in PAGE-EVENT. We hypothesize that this extra material causes spurious words to have high probability for the year, a problem avoided by the more focused SENTENCE-EVENT model.

Figure 2 shows the NDCG results for the queries with multiple relevant dates. In this case recall is slightly more important and having broad coverage of topics matters. The YEAR-BOOK-EVENT and SENTENCE-EVENT models continue to perform well on this task, but the YEAR-EVENT model roughly matches them.

In almost all cases, the proximity features included in the sequential dependence model (SDM) improved results, except in the many relevant year case with the SENTENCE-EVENT model, although those documents are so short that if two terms occur, they are already close, and the more general language of those queries might be causing problems.

Across both tasks, we find that YEAR-BOOK-EVENT and SENTENCE-EVENT perform the best overall, and that performance-wise, neither seems to have a clear advantage.

6.2 Considering Efficiency

One of the advantages of the YEAR-BOOK-EVENT is that it offers an interesting tradeoff not only in terms of effectiveness, but also in size. The inverted index for the YEAR-EVENT is the most efficient, occupying only 227 MiB on disk, whereas the postings for the SENTENCE-EVENT is over twice the size: 553 MiB. The 477 MiB of the YEAR-BOOK-EVENT postings provides a tradeoff between those extremes. While these three models were all built from the same context, we note here that fewer models with more context compress better on disk. Note that all of these models are small in comparison to the collection (90 GiB), but that efficiency may be a concern if used as an initial retrieval step or in conjunction with document retrieval over the whole corpus. This leads us to the ultimate conclusion that the YEAR-BOOK-EVENT model is preferable.

6.3 Revisiting the Raids on Deerfield

The motivating example in the introduction was a user interested in raids on the town of Deerfield, Massachusetts, during the colonial era. Revisiting this query directly on our best performing models gives us a concrete sense of the strengths and weaknesses of each, if only for one data point. Recall that the two relevant years are 1675 and 1704.

Issuing this query against the YEAR-EVENT model, we find 1704 at rank 2, and 1675 at rank 10. The rank 1 result is a false positive created by text in the margin next to a sentence summarizing the 1704 raid. This example, and the ones between ranks 2 and 10 demonstrate how even though the Year-Event might be an efficient and simple approach, it suffers when looking for specific events, as it is susceptible to more frequent mentions of Deerfield among other years.

The Sentence-Event approach is more robust, giving a topical result first: a 1713 attempt to recover prisoners, and then the 1704 raid itself in multiple appearances, with the 1675 raid buried under the several pages of results (it is only rarely mentioned in our corpus).

The results from the Year-Book-Event are more balanced. We still get 1713, 1704, and the false positive of 1602, but the 1675 raid appears at rank 10, after a series of mostly true positives, which suggests that of all the models, for this query, it was most able to balance the precision of avoiding the noise and breadth of the books corpus with the recall of still being able to retrieve rare events.

7 Conclusion

We present models for using historical data to predict the year of a query. Unlike past work in the newswire domain, find that that year modeling is too general for this task. The leading techniques are a nearest sentence approach and a joint year-document model. While the nearest sentence approach is generally as efficient as the more complicated model, it is more expensive in terms of space and in terms of the number of documents to rank. Altogether, we conclude that the Year-Book-Event model is prefered for this task.

In addition to our experimental results, we contribute an automatically collected set of over 40,000 queries tied to a single year. We also described a mechanism for creating merged "under-specified" queries with multiple years as their target. This dataset is publicly available at *(suppressed for review)*.

We believe that the promising results shown by our joint document and year event models suggest applicability for general entity models. The authors hope that work in this area will begin to unlock the possibilities of using the millions of digital books available online.

8 Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

1. O. Alonso, M. Gertz, and R. Baeza-Yates. On the value of temporal information in information retrieval. In *SIGIR 2007*, volume 41, pages 35–41. ACM, 2007.

2. O. Alonso, J. Strötgen, R. A. Baeza-Yates, and M. Gertz. Temporal information retrieval: Challenges and opportunities. *TWAW*, 11:1–8, 2011.
3. M. Brucato and D. Montesi. Metric spaces for temporal information retrieval. In *Advances in Information Retrieval*, pages 385–397. Springer, 2014.
4. R. Campos, G. Dias, A. Jorge, and C. Nunes. Gte: A distributional second-order co-occurrence approach to improve the identification of top relevant dates in web snippets. In *CIKM 2012*, New York, NY, USA, 2012. ACM.
5. H. T. Dang and K. Owczarzak. Overview of the TAC 2008 opinion question answering and summarization tasks. In *Proc. of the First Text Analysis Conference*, 2008.
6. M. Daoud and J. Huang. Exploiting temporal term specificity into a probabilistic ranking model. 2011.
7. F. de Jong, H. Rode, and D. Hiemstra. Temporal language models for the disclosure of historical text. Royal Netherlands Academy of Arts and Sciences, 2005.
8. J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.
9. K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR 2000*, pages 41–48. ACM, 2000.
10. A. Jatowt, C.-M. Au Yeung, and K. Tanaka. Estimating document focus time. In *CIKM '13*, New York, NY, USA, 2013. ACM.
11. N. Kanhabua and K. Nørvåg. A comparison of time-aware ranking methods. In *SIGIR 2011*, pages 1257–1258.
12. N. Kanhabua and K. Nørvåg. Improving temporal language models for determining time of non-timestamped documents. In *Research and Advanced Technology for Digital Libraries*, pages 358–370. Springer, 2008.
13. N. Kanhabua and K. Nørvåg. Determining time of queries for re-ranking search results. In *Research and Advanced Technology for Digital Libraries*, pages 261–272. Springer, 2010.
14. G. Kazai, M. Koolen, J. Kamps, A. Doucet, and M. Landoni. Overview of the INEX 2010 book track: Scaling up the evaluation using crowdsourcing. In *Comparative Evaluation of Focused Retrieval*, pages 98–117. Springer, 2011.
15. A. Kumar, J. Baldridge, M. Lease, and J. Ghosh. Dating texts without explicit temporal cues. *arXiv preprint arXiv:1211.2290*, 2012.
16. X. Li and W. B. Croft. Time-based language models. In *CIKM 2003*, pages 469–475. ACM, 2003.
17. C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.
18. D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR 2005*, pages 472–479. ACM, 2005.
19. D. Metzler, R. Jones, F. Peng, and R. Zhang. Improving search relevance for implicitly temporal queries. In *SIGIR 2009*, pages 700–701. ACM, 2009.
20. S. Nunes, C. Ribeiro, and G. David. Use of temporal expressions in web search. In *Advances in Information Retrieval*, pages 580–584. Springer, 2008.
21. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR 1998*, pages 275–281. ACM, 1998.
22. J. Pustejovsky, J. M. Castano, R. Ingria, R. Sauri, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34, 2003.

23. D. A. Smith. Detecting and browsing events in unstructured text. In *SIGIR 2002*, pages 73–80. ACM, 2002.
24. H. M. Sylvester. *Indian Wars of New England*, volume 2. 1910. https://archive.org/details/indianwarsneweng02sylvrich.
25. E. M. Voorhees et al. The TREC-8 Question Answering Track Report. In *TREC*, volume 99, pages 77–82, 1999.
26. C. Willis and M. Efron. Finding information in books: characteristics of full-text searches in a collection of 10 million books. 2013.