The Thesis Committee for Abhimanu Kumar
certifies that this is the approved version of the following thesis:

**Supervised Language Models for Temporal Resolution of Text in Absence of Explicit Temporal Cues**

APPROVED BY

SUPERVISING COMMITTEE:

Joydeep Ghosh, Supervisor

Jason Baldridge

Matthew Lease

# Supervised Language Models for Temporal Resolution of Text in Absence of Explicit Temporal Cues

**by**

**Abhimanu Kumar, B.Tech.**

**THESIS**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Master of Science in Computer Science**

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2013

Dedicated to my mother and father Madhu and Satyendra Shahi.

# Acknowledgments

I wish to thank all of my three mentors: Prof. Joydeep Ghosh, Prof. Jason Baldridge and Prof. Matthew Lease for their immense support, intellectual and otherwise without which my masters degree would not have been possible.

**Supervised Language Models for Temporal Resolution of Text in Absence of Explicit Temporal Cues**

Abhimanu Kumar, M.S.C.S.
The University of Texas at Austin, 2013

Supervisor: Joydeep Ghosh

This thesis explores the temporal analysis of text using the implicit temporal cues present in document. We consider the case when all explicit temporal expressions such as specific dates or years are removed from the text and a bag of words based approach is used for timestamp prediction for the text. A set of gold standard text documents with timestamps are used as the training set. We also predict time spans for Wikipedia biographies based on their text. We have training texts from 3800 BC to present day. We partition this timeline into equal sized chronons and build a probability histogram for a test document over this chronon sequence. The document is assigned to the chronon with the highest probability.

We use 2 approaches: 1) a generative language model with Bayesian priors, and 2) a KL divergence based model. To counter the sparsity in the documents and chronons we use 3 different smoothing techniques across models. We use 3 diverse datasets to test our models: 1) Wikipedia Biographies, 2) Guttenberg Short Stories, and 3) Wikipedia Years dataset.

Our models are trained on a subset of Wikipedia biographies. We concentrate on two prediction tasks: 1) time-stamp prediction for a generic text or mid-span prediction for a Wikipedia biography , and 2) life-span prediction for a Wikipedia biography. We achieve an f-score of 81.1% for life-span prediction task and a mean error of around 36 years for mid-span prediction for biographies from present day to 3800 BC. The best model gives a mean error of 18 years for publication date prediction for short stories that are uniformly distributed in the range 1700 AD to 2010 AD. Our models exploit the temporal distribution of text for associating time. Our error analysis reveals interesting properties about the models and datasets used.

We try to combine explicit temporal cues extracted from the document with its implicit cues and obtain combined prediction model. We show that a combination of the date-based predictions and language model divergence predictions is highly effective for this task: our best model obtains an f-score of 81.1% and the median error between actual and predicted life span midpoints is 6 years. This would be one of the emphasis for our future work.

The above analyses demonstrates that there are strong temporal cues within texts that can be exploited statistically for temporal predictions. We also create good benchmark datasets along the way for the research community to further explore this problem.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

This thesis deals with the assigning time to text: either a specific point of time on a timeline or a range depending upon the prediction task. We propose temporal analysis models that leverage ideas present in computational linguistics and information retrieval. While there have been prior research that focus on extracting explicit mentions of temporal expressions [2], in this work we primarily focus on the implicit temporal cues present in a text. This extraction of implicit cues present in the text relies on the temporal distribution of words in the corpora across time periods. By combining several such distributions across time and corpora we can infer a unique temporal distribution for a test document.

## 1.1 Research Motivation

Accurate extraction and resolution of explicit mentions of time (absolute or relative) is clearly important [2] and the problem becomes further complex if there are no explicit temporal cues present in the text. A system that can assign time to text with no explicit temporal cues can be used for various document dating applications.

In addition to document dating, such a system has potential to inform work in computational humanities, specifically scholars' understandings of how a work was influenced by or reflects different time periods. It can provide clues as to how specific words and terms in a language got into popular use and disuse over time. We can infer periodic events in a per-

sons life via words that have different periods of temporal behavior. For example, *breakfast* and *dinner* cycle every day, *weekend* cycles every week, *paycheck* cycles every month, and *winter* and *summer* cycle every year. Data is now available for them via constant newswire and Twitter feeds, and it is likely to be quite important to detect and tease apart such periodic properties of words, especially for more fine-grained temporal resolution.

This system can also be used to keep track of terminology changes over time. There are words in language that change their meaning or get obsolete over time. For example "Siam" was used as a name for Thailand in the early 20th century but not at all anymore. By obtaining the temporal distribution of "Siam" over a heterogeneous set of corpora we can infer that "Siam" is obsolete these days.

## 1.2 Research Problems

From the discussion in section 1.1 *it is our objective to assign time-stamps or time-intervals to test documents*. Providing a time-stamp to an incoming document involves three major challenges: 1) a competitive text model that can provide an accurate prediction using a training set, 2) smoothing for low-evidence training and test documents, and 3) analysis of the prediction models over diverse dataset to evaluate the robustness of the technique.

### 1.2.1 Prediction Model

Statistical prediction model have become prevalent in computational linguistics in recent years as they are more accurate and insightful compared to rule-based models. From divergence and language-modeling based techniques to regression and graphical models have provided great insights into text data. Among this diversity our aim is to find a statistical model that is most suitable for the temporal analysis problem. One of the research problems

that we want to solve is:

- *(RP1) What is the best statistical model for the time-stamp prediction task?*

We try to restrict the search space of our model to a subset of statistical techniques: divergence based models and language models to tackle the generic research problem (RP1) above.

### 1.2.2 Smoothing Techniques

Prediction models do not perform well when they have small evidence for a prediction task and use various smoothing techniques to counter it. A small text document has very little predictive content and thus needs to be augmented with evidence from the training set or a generic corpora. But different smoothing techniques have different strength (and weaknesses). It is important to pick the most suitable smoothing technique for a given model and dataset as it provides increased accuracy as well as unique insights into the dataset being dealt with. Hence the second research question is:

- *(RP2) How various smoothing techniques fare for a given model and dataset for prediction?*

Again we restrict our set of smoothing techniques to the more prevalent ones such as Dirichlet, Jellenick Mercer smoothing etc.

### 1.2.3 Heterogeneous Corpora

Given a document dating model, it is important to analyse the model's robustness. A model that is able to predict timestamps for a diverse set of documents is highly desirable.

We try to solve this research problem for our set of models. Specifically we determine:

- *(RP3) How our models fare on different test corpora?*

  We analyse this research question over 3 diverse set of test corpora.

## 1.3 The Approach

Our primary focus here is the challenge of learning implicit temporal distributions over natural language use at-large. As a concrete example, consider the word *wireless*. Introduced in the early 20th century to refer to radios, it fell into disuse and then its meaning shifted later that century to describe any form of communication without cables (e.g. Internet access). As such, the word *wireless* embodies implicit time cues, a notion we might generalize by inferring its complete temporal distribution as well as that of all other words. By harnessing many such implicit cues in combination across a document, we might further infer a unique temporal distribution for the overall document.

As in prior document dating studies, we partition the timeline (and document collection) to infer an unique language model (LM) underlying each time period [20, 22]. However, while prior work considered texts from the past 10-20 years, our work is far more historically-oriented, learning temporal distributions from the present day back to 3800 B.C. We also predict publication dates for historical works of fiction. Another point of distinction is that we learn from and predict time spans in addition to single time instants.

We accomplish this by learning from a different kind of text: Wikipedia biographies. By associating each individual's lifetime with the textual description of their lives, we learn word-time affinities for each lifespan. Wikipedia-based training is further advantageous since

its recency enables us to control against stylistic vs. content factors influencing vocabulary use (e.g. consider the difference between William Mavor's 1796 discussion[1] of Sir Walter Raleigh vs. a modern retrospective biography[2])

## 1.4 Contribution

Our primary contribution is to use the implicit cues present in text to assign timestamp. We propose 5 different models and evaluate them over 3 different datasets for timestamp prediction. We train the models over a set of Wikipedia biographies and predict for the other 2 datasets using the same learnt model. In this process we demonstrate transfer learning capabilities of the models. We predict an individual's mid-life-span from the text of his biography. Our best model achieves a median mid-life-span distance of around 22 years and a mean distance of 36 years. We also predict the individual's lifetime from the text of his biography. Our best model achieves an $f$-score of 81% for overlap with actual lifetimes.

A second task is to predict publication dates of short stories from the Gutenberg project.[3] In comparison to Wikipedia biographies, these stories use relatively few explicit temporal expressions or mentions of real-life named entities. We rely on the same time-based language models learned from Wikipedia to predict these publication dates. This difference between train vs. test genres, coupled with lack of explicit temporal cues, makes the task particularly challenging, but our best model still achieves a mean distance of around 20 years and a median distance of 17 year from the true publication date.

A third task is to predict the year of occurrence of a set of events. Wikipedia

[1] `http://bit.ly/lKR8Aa`
[2] `http://en.wikipedia.org/wiki/Walter_Raleigh`
[3] `http://www.gutenberg.org`

maintains a collection of year-wise events that occurred in the past [4]. Each document in this collection corresponds to a single year and each year has only one document in this collection that contains a list of events that happened in that year. We again rely on the same time-based language models learned from Wikipedia to predict the year of occurrence for a given page. The pages are from 500 B.C. to 2010 A.D. The best model gives mean error of 36 years and median error of 21 years.

## 1.5 Publications

During the course of this masters thesis we have the following works published or under submission:

[1] (**Accepted**) Abhimanu Kumar, Matt Lease, and Jason Baldridge. 2011. Supervised Language Modeling for Temporal Resolution of Texts. In Proceedings of the 20th ACM Conference on Information Knowledge and Management (CIKM). Glasgow, Scotland.

[2] (**Submitted**) Abhimanu Kumar, Matt Lease, Jason Baldridge, and Joydeep Ghosh. 2012. Effective Methods for Automatic Dating of Text in Absence of Explicit Temporal Cues. In Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP). Jeju, South Korea.

## 1.6 Thesis Organisation

In the 2nd chapter, we provide a brief background of prior work done in the area of temporal prediction. In chapters 3 the datasets used are described and in chapter 4 the models applied are discussed. Chapters 5 and 6 describe the evaluation metrics and the

[4] `http://en.wikipedia.org/wiki/List_of_years`

experimental setup, respectively. Chapter 7 analyses the results obtained as well as the interesting attributes of the models and datasets. Chapter 8 and 9 conclude this document by providing an overview of the contibutions of this work and further areas of improvement.

# Chapter 2

# Background and Related Work

Temporal analysis of text is an active area of research since the early days of text mining with different focus in different areas. While the computational linguistics research in the early days was primarily concerned with the fine-grained ordering of temporal events [1, 47], information retrieval research focus has been largely on time-sensitive document ranking [11, 31], temporal organization and presentation of search results [2], how queries and documents change over time [24], etc. *The main focus of this thesis is assigning timestamps to documents in absence of any explicit temporal cue.*

## 2.1 Annotation and corpora

Recent years have brought increased interest in creating new, richly annotated corpora for training and evaluating time-sensitive models. TimeBank [41] and Wikiwars [38] are great exemplars of such work. They have been used for tasks like modeling event structure (e.g. work of [9] on TimeBank).

## 2.2 Analyzing literary and social media

Another line of work pursued shallower analysis applied to historical and literary documents, as well as to microblogging data such as tweets and search queries. For example,

the Google N-Grams Viewer[1] allows word sequences to be plotted on a timeline with respect to their relative frequency per year, based on counts obtained from millions of books and their associated publication dates [16]. Time information in a historical or literary document gives an insight into how events happen or how topics vary over time. [17] provide a semantic approach to create and automatically update a database on infectious disease outbreaks. Time information in tweets also provide valuable information about current trendy topics in the microblogging world [54] crucial to enterprises reliant upon market sentiment.

## 2.3 Time-sensitive topic modeling

There has been a variety of work on time based topic-analysis in texts in recent years, such as Dynamic Topic Models [7]. Subsequent work [6] proposes probabilistic time series models to analyze the time evolution of topics in a large document collection. They take a sequential collection of documents of a particular area e.g. news articles and determine how topics evolve over time - topics appearing and disappearing, new topics emerging and older ones fading away. [49] provide a model to evaluate variations in the occurrence of topics in large corpora over a period of time. There have been other interesting contributions such as work by [36] which studies the history of ideas in a research field using topic models, by [10] which provides the temporal analysis of blogs and by [54] which gives models for mining cluster evaluation from time varying text corpora.

[1] `http://ngrams.googlelabs.com/`

## 2.4 IR Applications

IR research has investigated time-sensitivity query interpretation and document ranking [11, 31], time-based organization and presentation of search results [2], how queries and documents change over time [24], etc. One of the first LM-based temporal approaches by Li and Croft [31] used explicit document dates to estimate a more informative document prior. More recent work by Dakka et al. [11] automatically identify important time intervals likely to be of interest for a query and similarly integrate knowledge of document publication date into the ranking function. The most relevant work to ours is that by Alonso et al. [2], who provide valuable background on motivation, overview and discussion of temporal analysis in IR. Using explicit temporal metadata and expressions, they create *temporal document profiles* to cluster documents and create timelines for exploring search results.

## 2.5 Document dating

The most closely related work we are aware of studied LM-based document dating [20, 22]. We similarly partition the timeline (and document collection) to infer an unique LM underlying each time period. We also similarly estimate a LM underlying each document [40] and measure similarity between eac document's LM and each time period's LM, yielding a distribution over the timeline for each document. While prior work focused on the past 10-20 years, our work is far more historically-oriented, modeling the timeline from the present day back to 3800 B.C. We also predict publication dates for historical works of fiction, as well as learn from and predict time spans in addition to time instants.

The foundational work by de Jong et al. [20] considered Dutch newspaper articles from 1999-2005 and compared language models using the normalised log-likelihood ratio

measure (NLLR), a variant of KL-divergence. Linear and Dirichlet smoothing were applied, apparently to the partition LMs but not the document LMs. They also distinguish between output granularity of time (to be predicted) and the granularity of time modeled. Kanhabua et al. [22] extended de Jong et al.'s model with notions of temporal entropy, use of search term trends from Google Zeitgeist, and semantic pre-processing. Temporal entropy weights terms differently based on how well a term distinguishes between time partitions and how important a term is to a specific partition. Semantic techniques included part-of-speech tagging, collocation extraction, word sense disambiguation, and concept extraction. They created a time-labeled document collection by downloading web pages (mostly web versions of newspaper articles) from the Internet Archive which spanned roughly an eight year period. In follow-on work inferring temporal properties of queries [23], they used the New York Times annotated corpus, with articles spanning 1987-2007.

## 2.6 Geolocation

Temporal resolution can be seen as a natural pairing with geolocation: both are ways of connecting texts to simple, but tremendously intuitive and useful, models of aspects of the real world. There has been a long-standing interest in finding ways to connect documents to specific places on the earth, especially for geographic information retrieval [3, 14]. Of particular relevance to our paper is Wing and Baldridge's LM based method of measuring similarity of documents with language models for discrete geodesic cells on the earth's surface [51].

## 2.7 Authorship attribution

A final relevant work of note is the LM-based authorship attribution work by Zhao et al. [56]. They similarly partition the corpus by author, build partition-specific LMs, and infer authorship based on model similarity computed with KL-divergence and Dirichlet smoothing. They also consider English literature from the Gutenberg Project. Unlike us, they train directly on this corpus instead of applying LMs from another domain.

## 2.8 Computer vision

Divergence based techniques have been used in the area of computer vision to assign images to motion forms or event. Each standard form like the chronon has a distint distribution over the pixel set. A test image's distribution is compared to these standard distributions to assign similarity metrics. Khokhar et al. [42] use this to classify test images from different corpora to various motion events such as *left turn*, *right turn*, *convergence* etc. They estimate distributions as continuos than discrete assuming Gaussian form and use sampling techniques to evaluate KL divergence to avoid calculating the precise distribution form. They are able to avoid smoothing by assuming continous distibution over the pixel set though they have much bigger evidence set (training pixels) and can afford to estimate a continuous distribution without using smoothing. Saleemi et al. [18] use clustering techniques to temporally segment motion cues from a video and then use KL divergence to assign forms to test images and in the process avoid smapling techniques for divergences though clustering is more computational expensive in this case.

# Chapter 3

# Corpora

Our models are trained and evaluated on three datasets [1]

## 3.1 Wikipedia Biographies (wiki-bio)

The Wikipedia dump of English on September 4, 2010 is used [2] to extract biographies of individuals who lived between the years 3800 B.C. to 2010 A.D.

We extract the lifetime of each individual via each article's *Infobox* `birth_date` and `death_date` fields. We exclude biographies which do not specify both fields or which fall outside the year range considered. We treat the lifetime of each individual as the article's labeled time *span* and use these spans to train and evaluate our methods, e.g. predicting the mid-point of an individual's lifetime from his biography's text, or most likely year he was alive. As is often typical of Wikipedia coverage, the distribution of biographies is quite skewed toward recent times. 3.1 plots the number of birth per year in the training set.

We extract a total of 280,867 Wikipedia biographies of individuals whose lifetimes begin and end within the year range considered (3800 B.C. to 2010 A.D.). These biographies are split 80/10/10 into subsets for training (224,476 articles), development (28,212 articles)

[1] All the three will be released upon publication, including processing and extraction needed for easy replication of experiments.

[2] `http://download.wikimedia.org/enwiki/20100904/enwiki-20100904-pages-articles.xml.bz2`

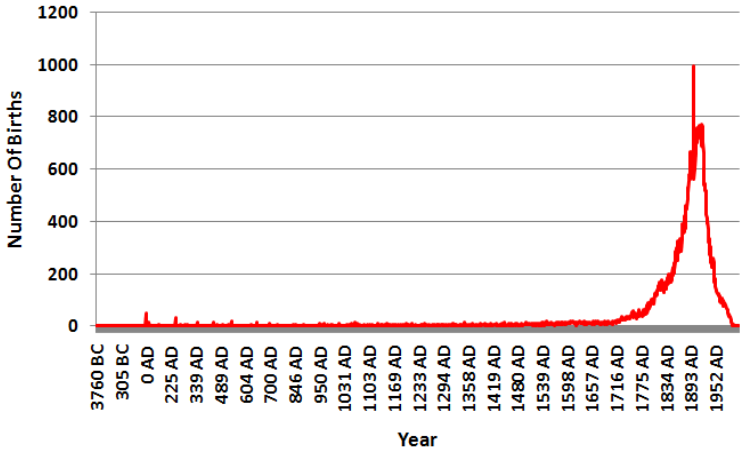



Figure 3.1: Graph of number of births per year in the Wikipedia biography training set.

and testing (28,179 articles). We remove all the documents from development and test set whose either of their `birth_date` or `death_date` missing. This gives use 8,358 articles in dev set and 8,440 articles in test set.

## 3.2 Gutenberg Short Stories (gutts)

We also consider a collection of English short stories, published between 1798 to 2008, obtained from the Gutenberg Project. Whereas with Wikipedia biographies we use labeled time spans corresponding to lifetimes, Gutenberg stories are labeled by publication year. Our inference task is then to predict the publication year given the story's text. We perform this task out-of-domain, applying models trained on the Wikipedia biographies to predict Gutenberg stories' publication dates. We use 678 Gutenberg short stories. The average, minimum and maximum word count of these stories are (roughly) 14,000, 11,000 and 100,000 respectively. Stories are split into a development set of 333 documents and a test set of 345 documents.

## 3.3 Wiki-Year Pages (wiki-year)

.

Wikipedia has a collection of pages corresponding to various years that describe the events that occurred for a given year. Each year has a corresponding page in Wikipedia and each page has a corresponding year in this collection [3]. This collection has pages starting form year 500 B.C. to 2010 A.D. Each page has the corresponding year as its label and the text contains all the events that occurred in that year. Our task is to predict the year of occurrence given the text. These texts are further filtered to remove any temporal expressions if present. The years are divided into even and odd sets. The even set is used for validation and odd set is used for test. We use 2,511 wiki-year documents. 1,256 documents become part of dev set and 1,255 are used for test set.

Figure 3.1 shows random sample lines from five wiki-year pages. The lines are terse and the text as a whole contain very little temporal expressions. The frequency of a typical word in this dataset is between 1 and 3 occurrences per document. Each event is a sort line in the text and sometimes events overlap among two documents temporally close to each other. This happens for the B.C. region of years as some events have a general know time period rather than a fixed year. For example the events "Proto-Greek invasions of Greece.", "Minoan Old Palace (Protopalatial) period starts in Crete." etc. are present in the text for 1878 as well as 1880 B.C. These occurred around 1880 B.C. but their exact occurrence date is unknown. We avoid such overlapping docuemnts and only use documents from 500 B.C. onwards till 2010 A.D.

[3] `http://en.wikipedia.org/wiki/List_of_years`

| Year | Sample Text |
|---|---|
| 1900 B.C. | Port of Lothal is abandoned.<br>Senwosret III (Twelfth Dynasty) started to rule.<br>Proto-Greek invasions of Greece.<br>Hittite empire in Hattusa, Anatolia.<br>Fall of last Sumerian dynasty. |
| 1000 B.C. | Early Horizon period starts in the Andes.<br>Iron is introduced in Ancient India.<br>Phoenician alphabet is invented.<br>Chavin culture starts in the Andes.<br>Paracas culture starts in the Andes. |
| 2 A.D. | Cedeides becomes Archon of Athens.<br>Deng Yu, Han Dynasty general and statesman.<br>Publius Alfenus Varus and Publius Vinicius become Roman Consuls.<br>Lucius Caesar, son of Marcus Vipsanius Agrippa and Julia the Elder, and heir to the throne.<br>The Chinese census shows nearly one million people living in Vietnam. |
| 1000 A.D. | Middle Horizon period ends in the Andes.<br>Dhaka, Bangladesh, is founded.<br>The Diocese of Koobrzeg is founded.<br>Stephen I becomes King of Hungary, which is established as a Christian kingdom.<br>Gunpowder is invented in China. |
| 2000 A.D. | Stipe Mesic is elected president of Croatia.<br>The Tate Modern Gallery opens in London.<br>Tuvalu joins the United Nations.<br>The last Mini is produced in Longbridge.<br>Tuanku Syed Sirajuddin becomes Raja of Perlis. |

Table 3.1: Sample text from 5 different years in wiki-year dataset.

## 3.4 Notation

.

With regards to notation and nomenclature, we refer to biographies, stories and Wiki-Year pages alike as *documents*, and each dataset as defining a document *collection* $c$ consisting of $N$ documents: $c = d_{1:N}$.

# Chapter 4

# Modeling and Estimation

This section describes the approach to represent time in our models and the 2 essential set of models that are utilised for 2 main tasks as described in previous sections.

## 4.1 Representing Time

Following aforementioned prior work [2, 20, 22], we quantize continuous time into discrete units. Our terminology and formalization most closely follow that of Alonso et al. [2]. The smallest temporal granularity we consider in this work is a single year, though the methods we describe can in principle be used with units of finer granularity such as days, weeks, months, etc.

### 4.1.1 Span

Let a *span* of multiple, contiguous years be some interval $\tau = [y_s, y_e]$, where $y_s$ and $y_e$ refer to start and end years, respectively. For example, individual lifetimes (as reported in Wikipedia biographies) we want to predict are expressed as spans. As noted in §3, we also know the year range covered by each document collection and restrict our overall timeline correspondingly to the span $\tau_o = [y_0, y_Y)$, covering a total of $y_Y - y_0 = \Delta$ years.

### 4.1.2 Chronon

A *chronon* is an atomic interval $x$ upon which a discrete timeline is constructed [2]. In this paper, a chronon consists of $\delta$ years, where $\delta$ is a tunable parameter. Given $\delta$, the timeline $T_\delta$ is decomposed into a sequence of $n$ contiguous, non-overlapping chronons $\mathbf{x} = x_{1:n}$, where $n = \frac{\Delta}{\delta}$.

### 4.1.3 Granule

A *granule* specifies a sequence of multiple contiguous chronons $\omega = x_{j:k}$ [2]. A granule therefore constitutes a kind of span. However, granules typically lack sufficient granularity to precisely match the actual labeled span $\tau_*^d$ for each document. A span $\tau_\delta$ can only be expressed as a granule if it is "chronon-aligned" on the timeline (i.e. $\tau_\delta = [y_0 + k_s\delta, y_0 + k_e\delta]$, with $\{k_s, k_e\} \in \mathbb{N}$). Consequenty, while some methods we describe infer a representative granule $\omega^d$ for each document $d$, other methods refine these granules to predict finer-grained spans $\tau^d$ for higher accuracy.

## 4.2 Pseudo-documents

We generate the "pseudo-document" $d^x$ for each chronon $x$ by concatenating all training documents whose labeled span overlaps $x$. For example, for a chronon size $\delta$ of 25 years, the biography of Abraham Lincoln (1809-1865) would be included in the pseudo-documents for each of the chronons representing 1800-1825, 1826-1850, and 1851-1875.

**Models.** Our models are divided into 2 sets dependending upon the task they perform: 1) models for time-stamp prediction, amd 2) models for time-span prediction. For Wikipedia biographies we predict their mid-life-span as well as life-span. For Guttenberg short stories

and wiki-years we predict time-stamp, either a publication year or event year.

## 4.3 Time-stamp models

The models here are used to assign a time-stamp to each test document.

### 4.3.1 Modeling Affinity

We model the affinity between each chronon $x$ and a document $d$ by estimating the discrete distribution $P(x|d)$. In the next section, we use $P(x|d)$ to infer affinity between $d$ and different chronons. The mid-point of (see section **??**) the most likely chronon is then returned as the predicted year by the model. We define two primary models for estimating $P(x|d)$. First, an LM-based approach for inferring affinity between chronon $x$ and document $d$ via a generative scheme of generating documents from chronons. The second approach estimates $P(d|x)$ based on a model of divergence between latent unigram distributions $P(w|d)$ and $P(w|x)$ [27]. In this work, we adopt a similar approach of modeling the likelihood of each chronon $x$ for a given document $d$.

### 4.3.2 Ranking by Model Comparison

Given the "pseudo-document" $d^x$ associated with each chronon $x$, we estimate chronon model $\Theta^x$ from $d^x$. In prior work by Kumar et al. [26], $P(x|d)$ is then estimated by computing the unnormalized likelihood of $x$ given $d$ through standard (inverse) KL-divergence and normalizing this likelihood over all chronons $x_{1:n}$:

$$P(x|d) = \frac{\mathcal{D}(\Theta^d||\Theta^x)^{-1}}{\sum_x \mathcal{D}(\Theta^d||\Theta^x)^{-1}} \tag{4.1}$$

While they do not show this, it is straighforward to see that their formulation is rank equivalent to the standard model comparison ranking with negative KL-divergence [20, 22]:

$$P(x|d) \propto \mathcal{D}(\Theta^d||\Theta^x)^{-1} \stackrel{rank}{=} -\mathcal{D}_{KL}(\Theta^d||\Theta^x) \tag{4.2}$$

Lafferty and Zhai showed such ranking is equivalent to generating the query (i.e. query-likelihood) assuming a uniform document prior and the query model being estimated by relative frequency (i.e. maximum likelihood) [28]. This means that for our task, provided we adopt a uniform prior over chronons and estimate the document model by relative frequency, KL-ranking and document-likelihood approaches will be rank equivalent.

### 4.3.3 Ranking by Document Likelihood

Recall that for our task, the document is the "query" for which we wish to rank chronons, hence document-likelihood here corresponds to the traditional query-likelihood approach. As discussed above, document-likelihood and model comparison approaches are rank equivalent if the document model is estimated by maximum likelihood and we assume a uniform prior over chronons. Just as informed document priors (e.g. PageRank or document length) inform traditional document ranking, an informed prior over chronons has potential to benefit our task as well.

Using Bayes Rule, we estimate $P(x|d) \propto P(d|x)P(x)$. Assuming unigram modeling, the likelihood is given by:

$$P(d|x) = \prod_{w \in d^x} \theta_w^x \tag{4.3}$$

where $w$ is a word token in the pseudo-document $d^x$, and parameters of the chronon model $\Theta^x$ are estimated from pseudo-document $d^x$ as described in Section 4.3.4. We adopt a

chronon prior intuitively informed by the distribution of training documents over chronons:

$$P(x) = \frac{|d_{train} \in d^x|}{\sum_{\forall y} |d_{train} \in d^y|} \tag{4.4}$$

where $d_{train}$ is a training document, $d^x$ is the pseudo-document for chronon $x$ and $|d_{train} \in d^x|$ is the number of dated training documents overlapping chronon $x$.

### 4.3.4 Smoothing

We use three smoothing techniques: a) Jelinek-Mercer smoothing (JM) [53], b) Dirichlet smoothing [53], and c) chronon-specific smoothing (CS). For all three, for each word $w$, $\hat{\theta_w^d}$ can be computed as a mixture of document $d$ and document collection $c$ maximum-likelihood (ML) estimates:

$$\hat{\theta_w^d} = \lambda \frac{f_w^d}{|d|} + (1 - \lambda) \frac{f_w^c}{|c|} \ , \tag{4.5}$$

where $f_w^d$ and $f_w^c$ denote the frequency of word $w$ in the document or collection respectively, $|d|$ and $|c|$ are the document and collection lengths, and the parameter $\lambda$ specifies the smoothing strength.

With Dirichlet smoothing, we have:

$$\lambda = \frac{|d|}{|d| + \mu} \tag{4.6}$$

where $\mu$ is a hyper-parameter learned over validation set.

Chronon-specific smoothing is a special case of Dirichlet smoothing:

$$\mu = \frac{\xi}{|V_{d^x} \cup V_{d_i}|} \tag{4.7}$$

where $|V_{d^x} \cup V_d|$ denotes the document-chronon specific vocabulary for some collection document $d_i$ and pseudo-document $d^x$ and $\xi$ is a prior for hyper-parameter $\mu$ learned over validation set. For Jelinek-Mercer smoothing $\lambda$ is learned directly over validation set.

The intuition behind chronon-specific smoothing is that mass is provided only for the words that are present in the collection document or the chronon's pseudo-document, ignoring other words. Thus, the divergence calculation is done only with respect to either the document or the chronon we have evidence for. There are many chronons for which we have little textual evidence (e.g. those in the range before 500 B.C. and between 100 A.D. to 1600 A.D.); if these are smoothed with respect to all words in the collection, then those terms dominate the divergence calculation. When a short document is evaluated against a low-evidence chronon, smoothing over all words leads to many terms (few of which actually occur in the document or the chronon) having similar probabilities, leading to low divergence.

### 4.3.5 Inference

Section §4.3 described several models for estimating $P(x|d)$. From $P(x|d)$ we estimate the midpoint $\hat{y}$ or for each chronon. Every chronon $x = [y, y + \delta]$ is represented by its mid-point $\hat{y}$ where $\hat{y} = y + \delta/2$. In case of Wikipedia biographies the predicted $\hat{y}$ represents the mid-life-span of the individual, for Guttenberg short stories it is the publication date and for Wiki-Years dataset it is the year of happening of the given events page. In later sections we will present the baseline year-prediction for $\hat{y}$.

## 4.4 Time Span Models

The models here are used to assign a time span to each incoming test document. This is used for predicting life span of individula using their Wikipedia biographies.

### 4.4.1 WordAffinity Model

In this work, we adopt a similar approach to modeling the likelihood of each chronon $x$ for a given document $d$. By forming a unique "pseudo-document" $d^x$ associated with each chronon $x$, we estimate $\Theta^x$ from $d^x$ and estimate $P(x|d)$ by comparing the similarity of $\Theta^d$ and $\Theta^x$ [20, 22]. We compute the unnormalized likelihood of some $x$ given $d$ via standard (inverse) KL-divergence $\mathcal{D}(\Theta^d||\Theta^x)^{-1}$ as in section 4.3.2 and normalize in straight-forward fashion over all chronons $x_{1:n}$:

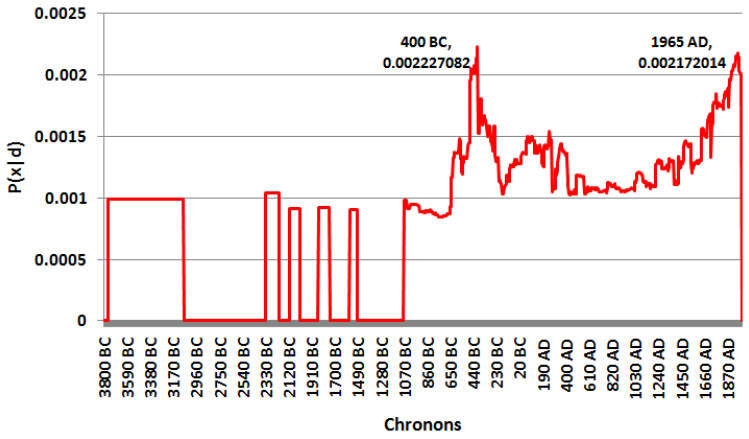
$$P_{wa}(x|d) = \frac{\mathcal{D}(\Theta^d||\Theta^x)^{-1}}{\sum_x \mathcal{D}(\Theta^d||\Theta^x)^{-1}} \tag{4.8}$$

**Smoothing** We use chronon-specific smoothing as defined in 4.3.4

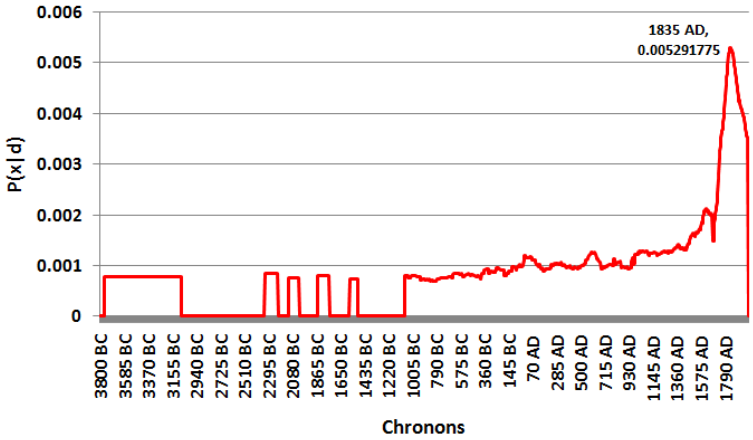
**Example distributions.** As examples of the kinds of distributions we obtain for documents in the development set, figure 4.1 shows graphs of $P_{wa}(x|d)$ for (a) Plato and (b) Abraham Lincoln. Recall that these are based on no explicit temporal expressions. For Plato, there is a clear spike around the time he was alive, along with another rising bump toward the current day, reflecting modern interest in him. For Lincoln, there is a single peak at 1835—very close to the 1837 midpoint of his life.

### 4.4.2 YearCounts model

YearCounts uses direct evidence of a document $d$'s temporality via explicit year mentions in the document. Essentially, this creates and uses a temporal document profile like that of [2], but it does not require existing named-entity recognizers and only considers years. This makes it both more limited yet also more lightweight, since it does not require training a temporal expression identifier.




(a)




(b)

Figure 4.1: Example $P_{wa}(x|d)$ distributions for the biographies of (a) Plato (428-348 B.C.) and (b) Abraham Lincoln (1809-1965).

**The model**. Let $f_y^d$ denote the frequency of year $y$ mentions found in $d$, and let $f^d = \sum_y f_y^d$ denote the total number of year mentions in $d$. We bin years by the chronons and then normalize over the timeline of all chronons to compute relative frequency: $P_{yc}(x|d) = \frac{1}{f^d} \sum_{y \in x} f_y^d$.

**Identifying year mentions**. We identify mentions of years via regular expressions (tuned on development data). For spans expressed in simple numerical form (e.g. "1973-1979"), we extract a mention for each year in the span. More complex expressions of spans result in our detecting only the start and end years. As this simple example shows, accurately identifying year mentions is non-trivial. To further illustrate, we list below examples of false year mentions identified by our initial set of regular expressions:

- **Month dates**: February 12, 12 February, Feb 12 etc. Here 12 can be picked up as an year if not removed
- **Ordinal Numbers**: of age 12, 2nd position, No. 2, #2, 3rd year
- **Amounts**: monetary information, decimal numbers, percentages: $200 million, 13.45, 15%
- **Miscellaneous**: 230px, 13:54, 134.jpg

Numeric values such as those in *"He received 456 votes out of 1347"* are difficult to differentiate from a valid year using regular expression since they do not have a fixed pattern. Thus, even after filtering out considerable noise, some tokens still are identified erroneously as valid years which skews our estimation. Most of the intractable noise seems to occur in range of 1-200.

Regardless of noise, YearCounts has two obvious limitations: (1) it ignores information from other tokens in the document $d$, and (2) it completely fails if there are few or no year mentions in $d$.

**Example distributions.** Figure 4.1 shows example distributions of $P_{yc}(x|d)$ for (a) Plato and (b) Abraham Lincoln. Note in particular that there is a great deal of noise in

the distribution for Plato in the first century A.D. The word model in figure 4.1(a) effectively balances that noise. Though the distribution for Lincoln is less affected by such noise, it does spike at 1860 due to the heavy focus on the American Civil War in his biography. The word model again balances this out so that the span of Lincoln's life can be better estimated in the combined model we describe next.

### 4.4.3 YW-Combined model

WordAffinity is not susceptible to the noise that profoundly affects YearCounts (and especially in in the lower range of 1-200), and it uses all of the textual evidence in the document. However, the distribution $P_{wa}(x|d)$ tends to be much smoother than $P_{yc}(x|d)$ and uses explicit year mentions much less directly, suggesting that a combination of the two complementary approaches may be most effective. We use simple linear interpolation for this:

$$P_{yw}(x|d) = \alpha P_{wa}(x|d) + (1-\alpha)P_{yc}(x|d) \tag{4.9}$$

where $\alpha$ is tuned on development data.

### 4.4.4 Inference

We now describe how we use $P(x|d)$ to infer a representative chronon $x^d$, granule $\omega^d$, and/or span $\tau^d$ for each document. We define six methods: MaxChronon, MaxGranule, CenterGranule, GranuleMaxChronon, TrimmedSpan, and VarianceSpan. The first two methods infer these time units directly, and the latter four methods utilize the output of the first two. We will refer to the most-likely chronon for a given granule as that

granule's *anchor*.

### 4.4.5 Chronon and Granule Identification

**MaxChronon** selects the most-likely chronon under $P(x|d)$:

$$\mathbf{x^d_{mc}} = \arg\max_{x \in \mathbf{x}} P(x|d) \tag{4.10}$$

The auxiliary function **CenterGranule** takes such a chronon $x^d = x_j$ as input and produces a representative granule $\omega^d$ for $d$ by centering a span of $l = k\delta$ years around $x^d$, where $k \in \mathbb{N}$ and $l$ is a parameter. If $(l - \delta)/\delta$ is odd, we will have one more chronon to the left of $x^d$ than to the right of it. For example, if $l = 120$ and $\delta = 5$, CenterGranule will produce granule $\omega^d_{mc} = x_{j-12:j+11}$ having 12 chronons left of $x^d$ and 11 chronons to its right. We define $\omega^{\mathbf{d}}_{\mathbf{mc}} = \text{CenterGranule}(x^d_{mc})$.

**MaxGranule** selects the most-likely granule $\omega^d_{mg}$ for $d$ according to $P(x|d)$. The size of this granule is a function of the parameter $l$: $\lfloor \frac{l}{\delta} \rfloor = w$ chronons. For example, if $l = 120$ and $\delta = 40$, MaxGranule considers granules of size $w = 3$ chronons. For the overall timeline of $n$ chronons, there are $n-w+1$ such granules $\omega = \omega_{1:n-w+1}$. We effectively slide a window of $w$ chronons across $\mathbf{x}$ one at a time, where each position corresponds to a unique granule. We then select the maximal granule:

$$\omega^{\mathbf{d}}_{\mathbf{mg}} = \arg\max_{\omega} \sum_{x \in \omega} P(x|d) \tag{4.11}$$

The probability of each consecutive granule can be efficiently computed since to move from one granule to the next we just remove the left-most chronon and add a new one on the right side.

Another auxiliary function, **GranuleMaxChronon**, takes an input granule $\omega^d$ and outputs its most-likely chronon $x \in \omega^d$:

$$x^d = \arg\max_{x \in \omega^d} P(x|d) \tag{4.12}$$

We define $\mathbf{x^d_{mg}} = \textsc{GranuleMaxChronon}(\omega^d_{mg})$. Whereas $x^d_{mc}$ corresponds to the most-likely chronon over all $\mathbf{x}$, $x^d_{mg}$ corresponds to maximal chronon within the most-likely granule $\omega^d_{mg}$.

Finally, we note two possible approaches not used. By construction, $\textsc{GranuleMaxChronon}(\omega^d_{mc}) = x^d_{mc}$ and so is redundant with MaxChronon. We also do not use $\textsc{CenterGranule}(x^d_{mg})$, which would center an $l$-sized granule around $x^d_{mg}$.

### 4.4.6 Span identification

So far we have two methods of inferring a representative granule $\omega^d$ for document $d$, yielding $\omega^d_{mg}$ and $\omega^d_{mc}$. While we could directly use either granule as a representative span $\tau^d$ for $d$ (since a granule is a kind of span), this has two limitations. First, both $\omega^d_{mg}$ and $\omega^d_{mc}$ consist of a fixed number of chronons regardless of the the shape of $P(x|d)$. In practice, the same size of granule may not be optimal for all $d$. Secondly, granules may not have sufficient granularity to match the optimal span $\tau^d_*$ for $d$, meaning any choice of granule would guarantee some loss of accuracy in attempting to precisely match actual document spans.

To address these limitations, we define two additional methods, TrimmedSpan and VarianceSpan, which each take a granule $\omega^d$ as input. TrimmedSpan addresses the first limitation by trimming one or more chronons from the left and right extent of $\omega^d$ to create a narrower granule $\hat{\omega^d}$. (We do not consider extending an input granule with additional

chronons.) VarianceSpan inspects the standard deviation of probability mass around $\omega^d$'s anchor chronon $x^d$ to generate a new span $\tau^d$ centered on $x^d$.

Specifically, **TrimmedSpan** trims the edges of $\omega^d = x_{m:n}$ with a simple heuristic based on thresholding the relative probability from one interval to the next. Consider the left-most chronon $x_m$, letting $x^l = x_m$. Given parameter $\kappa$, while the inequality

$$\frac{P(x^{l+1}|d) - P(x^l|d)}{P(x|d)} > \kappa \tag{4.13}$$

holds, we trim $x^l$ from $\omega^d$, redefine $x^l = x_{m+1}$ and $\omega^d = x_{m+1:n}$, and repeat. Trimming terminates once the inequality no longer holds or we have $|\omega^d| = 1$ (i.e. $\omega$ consists of a single chronon). Right-trimming proceeds similarly from the right- side rather than the left. The final trimmed granule is denoted $\hat{\omega^d}$.

**VarianceSpan** calculates the standard deviation of probability mass around the anchor chronon $x^d$ of input granule $\omega^d$. Let $\omega_l^d$ denote the sequence of chronons left of $x_j$ in $\omega^d$ (excluding $x^d$), and $|\omega_l^d|$ the number of chronons in $\omega_l^d$. Let $\bar{P}_l$ denote the simple average probability of chronons in $\omega_l^d$: $\bar{P}_l = \frac{1}{|\omega_l^d|} \sum_{x \in \omega_l^d} P(x|d)$. We the compute the standard deviation for $\omega_l^d$ by:

$$\sigma_l = \sqrt{\sum_{x \in \omega_l^d} ||P(x|d) - \bar{P}_l||^2} \tag{4.14}$$

We compute $\sigma_r$ similarly for $\omega_r^d$, the chronons right of $x^d$ in $\omega^d$. By computing the left and right standard deviations separately, we are able to capture some of the natural skew that dominates many biographies, wherein most of the (temporal) interest is nearer to an individual's death than to their birth (giving a higher s.d. on the left side than the right).

Let $\xi_s$ denote the first year of $x^d$. We use these standard deviation statistics to

generate the predicted span as:

$$\hat{\tau^d} = [\ \ \xi_s - \lfloor \gamma\sigma_l \rfloor,\ \xi_s + \lfloor \gamma\sigma_r \rfloor\ \ ] \tag{4.15}$$

where parameter $\gamma \in \mathbb{R}+$ denotes a stretch multiplier we tune on development data.

# Chapter 5

# Evaluation Metrics

This section describes the metrics used to eavluate our models.

## 5.1 Data Cleaning

We clean the documents to remove all temporal expressions (explicit or implicit dates/times) using Heidel-Time temporal tagger [44]. Heidel-Time tags all temporal expresion present in the text and we remove all such tagged words to avoid using any explicit temporal cues. Heidel-Time also provides the first two dates present in the text which is a very effective baseline for predicting the mid-life span of biographies. All numeric tokens and standard stopwords are removed. This gives a vocabulary size of 374,973 words for the entire Wikipedia biography corpus.

## 5.2 Metrics

We have two sets of evaluation metrics: 1) for time span prediction, and 2) for time-stamp prediction. These are described bellow in sepearte sections.

### 5.2.1 Time-stamp Metrics

**Parameters and Estimation**. For each model+task, we tune the parameters $\delta$, $\mu$, $\xi$, and $\lambda$ over the dev sets of the corresponding dataset (task).

As in prior work [20, 22, 26], we smooth chronon pseudo-document language models (for all models as well as smoothing techniques) but not document models. While smoothing both may potentially help, smoothing the former is strictly necessary to prevent division by zero in the KL-divergence calculation.

**Year Prediction**. For Wikipedia biographies, we take the midpoint of individual's lifetime as the gold standard year to match for the text. With Gutenberg stories, the gold standard to match for each story is a labeled year (publication date). For wiki-year the gold-standard is the year of the document.

**Error Measurement.** When predicting a single year for a document, a natural error measure between the predicted year $\hat{y}$ (mid-point) and the actual year $y^*$ is the difference $|\hat{y} - y^*|$. We compute this difference for each document, then compute and report the mean $\bar{y}$ and median $\tilde{y}$ of differences across documents. Similar distance error measures have also been used with document geolocation [15, 51].

**Baselines** For Wikipedia biographies the baseline is the mid-point of the first two temporal-dates extracted by Heidel-Time [44]. This is a highly efficient baseline as for Wikipedia biographies generally the first two dates are the `birth_date` and `death_date`. For Gutenberg stories, we take 1903, the midpoint of the range of publication dates (1798-2008) as the baseline. For Wiki-Year dataset the baseline is the midpoint of the prediction range i.e. $\frac{-500+2010}{2} = 755$.

### 5.2.2 Time Span Metrics

**Parameters and Estimation**. The models and methods we have described involve various parameters. The $l$ parameter of CenterGranule and MaxGranule is

fixed at 120 without tuning (since the longest document human lifespan is 122 years). All other parameters are tuned on the Wikipedia development data: the chronon size $\delta$, WordAffinity's smoothing parameter $\lambda$, the weighting parameter $\alpha$ for YW-Combined, TrimmedSpan's $\kappa$ threshold, and VarianceSpan's $\gamma$ parameter.

Here again we smooth chronon pseudo-document language models but not document models.

**Span Prediction**. We measure precision $\boldsymbol{P}$, recall $\boldsymbol{R}$, and F-score $\boldsymbol{F}$ of predicted spans $\hat{\tau}$ with respect to gold spans $\tau^*$ using years as the unit of measure. For example, predicting $\hat{\tau}$=1822-1874 ($|\hat{\tau}| = 52$ years) for $\tau^*$=1831-1889 ($|\tau^*| = 58$ years, with $|\hat{\tau} \cap \tau^*| =$ 43 years correctly matched) obtains $P = 82.7\%$ (43/52), $R = 74.1\%$ (43/58), and $F \approx 79\%$ ($\frac{2PR}{P+R}$). The values we report are the averages over all the documents being evaluated, given by ($\overline{\boldsymbol{P}} = \frac{\mathbf{1}}{|\mathbf{D}|}\sum_{\mathbf{d}\in\boldsymbol{D}}\mathbf{P}_{\boldsymbol{d}}$) and ($\overline{\boldsymbol{R}} = \frac{\mathbf{1}}{|\mathbf{D}|}\sum_{\mathbf{d}\in\boldsymbol{D}}\mathbf{R}_{\boldsymbol{d}}$) respectively for each model where $\mathbf{d}$ is a Wikipedia biography article and $\boldsymbol{D}$ is the set of biographies being evaluated.

A limitation of $\boldsymbol{P}$ and $\boldsymbol{R}$ is that zero credit is awarded no matter how closely a prediction missed the actual interval. Consequently, we also report aggregate mean $\bar{y}$ and median $\tilde{y}$ difference in years between the prediction and the gold standard to see how close we are to the actual value even if we miss it. Whereas with year prediction we use the first year of the predicted chronon, here we use the midpoint of the predicted span as the predicted year. We show that this in fact is a more accurate predictor (see §6.2.2).

Finally, the last metric used is $PR_0$ which gives the number of documents the model missed completely (for which P=R=0). This metric highlights situations where a model might make many high quality predictions and then be wildly off on others, e.g. in

the case of YearCounts when noise in the year extraction overcomes it and leads it to pick dates in the 0-200 A.D. range.

# Chapter 6

# Experiments

## 6.1 Time-stamp Prediction

### 6.1.1 Parameter Tuning

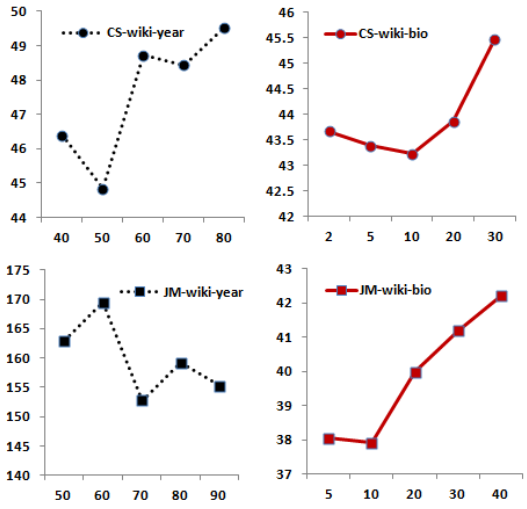
We begin with year prediction experiments on the development sets to tune the parameters $\delta$, $\xi$ or $\mu$. We parametrize $\mu$ as a function of the average chronon size:

$$\mu = \lceil \rho \bar{c} \rceil \tag{6.1}$$

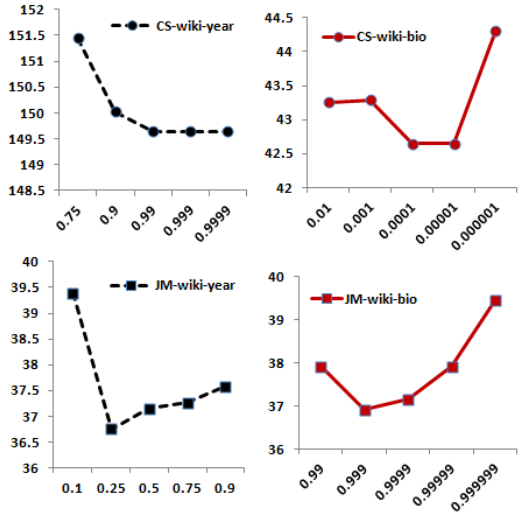
where $\bar{c}$ is the average chronon size in the training set. This is a constant whose value is dependent upon the model and the task. The value of $\rho$ is tuned over the validation set.

**Choice of chronon size and smoothing parameters.** We tune the chronon size ($\delta$) over the validation set and tune the smoothing parameters $\lambda$, $\rho$, and $\xi$ (depending on the type of smoothing) for the best $\delta$ obtained. For $\delta$ tuning we assign an arbitrary value to the smoothing parameter $\lambda$. The $\delta$ is tuned for each dataset and KL model with CS and JM smoothings. Bayesian model with Dirichlet/JM smoothing and KL model with Dirichlet smoothing use the same best $\delta$ obtained for KL model with JM smoothing on the respective datasets. Each dataset, model and smoothing triad tunes its own unique smoothing parameter $\lambda$, $\xi$, or $\rho$. The tuning parameters are trained on the mean-error and over 8,358 documents for Wikipedia biographies, 333 documents for Guttenberg Stories

and 1956 documents for wiki-years. Our search space for smoothing parameters $\xi$, $\lambda$ and $\rho$ includes $\{ 1e-12, 1e-11, \ldots, 0.1, 0.25, 0.75, 0.9, 0.99, \ldots, 0.999999999 \}$




(a)




(b)

Figure 6.1: Tuning for $\delta$ (fig. a) and smoothing parameters (fig. b) over wiki-bio and gutts datasets for KL model. The smoothing parameter $\xi$ (for CS smoothing) and $\lambda$ (for JM smoothing) are fixed at 0.01 and 0.99 respectively for $\delta$ tuning (fig a.)

| Dataset | Model | Smoothing | $\delta$ | $\bar{y}$ | $\tilde{y}$ |
|---|---|---|---|---|---|
| baseline | | | | 298.22 | 0.0 |
| wiki-bio | KL | $\xi$=$10^{-4}$ | 10 | 42.65 | 22.00 |
| wiki-bio | KL | $\lambda$=0.999 | 10 | 36.91 | 22.50 |
| wiki-bio | KL | $\rho$=$10^{-6}$ | 10 | 37.92 | 22.00 |
| wiki-bio | Bayesian | $\lambda$=0.999 | 10 | 36.65 | 22.00 |
| wiki-bio | Bayesian | $\rho$=$10^{-6}$ | 10 | 37.94 | 22.00 |
| baseline | | | | 37 | 50 |
| gutts | KL | $\xi$=$10^{-3}$ | 30 | 23.95 | 17.00 |
| gutts | KL | $\lambda$=0.999 | 10 | 20.41 | 17.00 |
| gutts | KL | $\rho$=$10^{-6}$ | 10 | 23.00 | 19.00 |
| gutts | Bayesian | $\lambda$=0.999 | 10 | 20.41 | 17.00 |
| gutts | Bayesian | $\rho$=$10^{-6}$ | 10 | 22.97 | 19.00 |
| baseline | | | | 978 | 489 |
| wiki-year | KL | $\xi$=0.99 | 70 | 149.64 | 29.00 |
| wiki-year | KL | $\lambda$=0.25 | 50 | 36.76 | 21.00 |
| wiki-year | KL | $\rho$=0.01 | 50 | 67.27 | 21.00 |
| wiki-year | Bayesian | $\lambda$=0.50 | 50 | 37.78 | 21.00 |
| wiki-year | Bayesian | $\rho$=0.01 | 50 | 65.39 | 21.00 |

Table 6.1: Model Results on dev set. JM=Jelinek-Mercer and CS=chronon-specific, BM=Bayes Model, and non-U=non-uniform prior

Figure 6.1 shows the tuning of $\delta$ and smoothing parameters ($\lambda$ for JM and $\xi$ for CS) for the wiki-bio and wiki-years dataset. All triplets formed by KL/Bayes model $\times$ JM/Dirichlet smoothing $\times$ wiki-years/wiki-bio/gutts dataset use the optimum chronon-size obtained for the respective datasets from the KL model with JM smoothing. Table 6.1 provides the best results for each triplet (model, data, smoothing) and the baseline for each dataset using the above described tuning scheme on the validation set. KL model with JM smoothing stands out to be the best across all the there datasets over the validation set.

From Figure 6.1 the mean error curve is generally smooth for $\lambda$ and $\xi$ unlike the $\delta$ (chronon-size parameter). This makes smoothing the LMs robust to a range of values. The $\delta$ has more fluctuation even in the optimal neighborhood, and this makes tuning chronon-size more intensive.

As noted in Section 4.3.2, model comparison (i.e. KL divergence) is rank-equivalent to Document-Likelihood assuming a uniform prior over chronons and that the document model is estimated by relative frequency [28]. In this work, we estimate our prior $P(x)$ from the training set such that the chronon with the higher number of documents gets higher probability $P(x)$ (equation 4.4). Another way to choose the prior $P(x)$ would be to assign it the Maximum Likelihood of the chronon $x$ in the chronon collection $X_{1..n}$. Such exploration of alternative priors will be investigated in our future work.

### 6.1.2 Test Results

| Dataset | Model | Smoothing | $\bar{y}$ | $\tilde{y}$ |
|---|---|---|---|---|
| baseline | | | 306.6 | 0.0 |
| wiki-bio | KL | $\xi$ | 42.8 | 22.5 |
| wiki-bio | KL | $\lambda$ | 37.4 | 22.5 |
| wiki-bio | KL | $\rho$ | 38.1 | 22.0 |
| wiki-bio | Bayesian | $\lambda$ | 37.3 | 22.5 |
| wiki-bio | Bayesian | $\rho$ | 38.0 | 22.0 |
| baseline | | | 37 | 50 |
| gutts | KL | $\xi$ | 39.6 | 19.0 |
| gutts | KL | $\lambda$ | 22.9 | 19.0 |
| gutts | KL | $\rho$ | 37.3 | 22.0 |
| gutts | Bayesian | $\lambda$ | 22.9 | 19.0 |
| gutts | Bayesian | $\rho$ | 37.4 | 23.0 |
| baseline | | | 978 | 489 |
| wiki-year | KL | $\xi$ | 143.6 | 30.0 |
| wiki-year | KL | $\lambda$ | 37.9 | 20.0 |
| wiki-year | KL | $\rho$ | 60.6 | 22.0 |
| wiki-year | Bayesian | $\lambda$ | 37.9 | 20.0 |
| wiki-year | Bayesian | $\rho$ | 52.1 | 20.0 |

Table 6.2: Model Results on test set. JM=Jelinek-Mercer and CS=chronon-specific, BM=Bayes Model, and non-U=non-uniform prior

**Wiki-bio year prediction.** Table 6.2 shows the results for the wiki-bio prediction on the wiki-bio test set (8440 documents). All the models beat the baseline with a huge

margin. Note that the baseline is not weak; it gives a median error of zero that means that it is correct atleast half of the time. The best model achieves a mean error of 37.35 years even though the prediciton range is 5810 (3800 B.C. to 2010 A.D.) years and the number of documents is 8440. Bayesian + JM smoothing provides the best result. Though the best median and mean error are not achieved by the same model. Its value of 22.0 means that the model is correct by atleast 22.0 years for atleast half of the documents. The mean and median error was 36.6 and 22.0 years for the best preforming model (Bayesian + JM smoothing) over dev set.

**Gutts prediction.** Table 6.2 shows the results for 345 short stories in the gutts test set. The stories are in the range 1798 to 2000 and the baseline is the mid-point of this range. There are very few explicit temporal expressions in the short stories so the baseline relies on the short range rather than any temporal expression present in the text (as was the case for wiki-bio baseline). This assumes that one knows a rough range of possible publication dates, which might be reasonable in many applications and provides a strong point of comparison. Recall that the model is trained on Wikipedia, so this both evaluates how well the model works on texts with very few explicit temporal expressions and how well it works on a different domain. All the models beat the baseline and the best model gives a mean error of 22.89 years and median error of 19 years. This means that the best model is off the true publication date by at most 19 years for no more than half of all the stories. For dev set, the mean and median error was 20.4 and 17.0 years for the best preforming model (Bayesian + JM smoothing).

**Wiki-years prediction.** The baseline is year 755, the mid-point of the range being predicted. All models beat the baseline comfortably. The best performing models (KL+JM

as well as Bayesian+JM) provide a mean error of 37.92 years and a median error of 20.0. This means that the prediction for atleast half of the documents is off by less than 20 years. KL model with JM smoothing provided the best mean and median error of 36.7 and 21.0 years respectively.

The standard smoothing techniques (JM and Dirichlet) perform better than the CS smoothing accross models and datasets. Among the conventional smoothings, JM performs slightly better than Dirichlet given the dataset and model. From Table 6.2 the median error for wiki-bio baseline is zero which is not surprising. For Wikipedia biographies, the first two dates present are birth and death dates in order with very high probability. And once the baseline picks it up its error goes to zero. This happens for atleast more than half of the documents being predicted which results in the median error as zero.

## 6.2 Time Span Prediction

The span predicion is done for Wikipedia biography set only.

### 6.2.1 Parameter tuning: year prediction

We begin with year prediction experiments on the development sets to tune the parameters $\delta$ and $\alpha$ and identify which of MaxChronon and MaxGranule is more effective for each model.

**Choice of chronon size and combined model weight.** The most important parameter for computing and using the interval histograms is the chronon size $\delta$. We tune it by evaluating the mean and median differences on 1000 documents from the Wikipedia development set. We also must pick the relative model weight $\alpha$ for YW-Combined. We

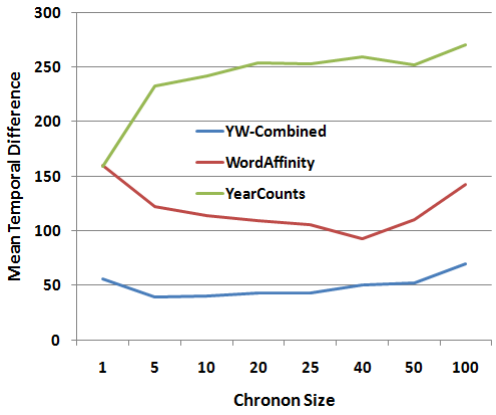
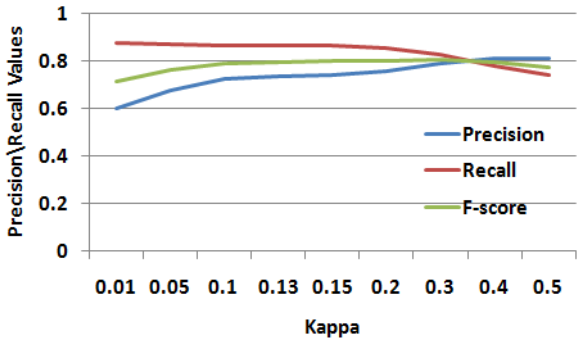


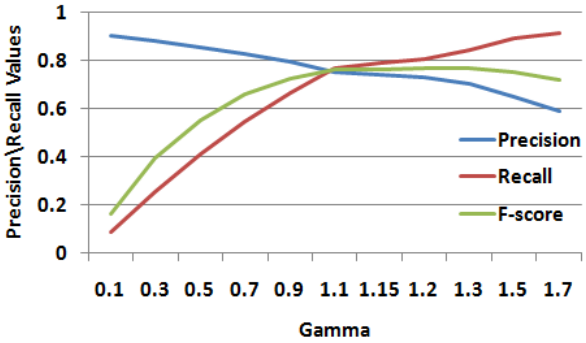

Figure 6.2: Mean temporal difference for different $\delta$ values for the three models.

do this by calculating the mean temporal difference for each model for $\delta$ in the range 1 to 100 and for $\alpha$ in the range .01 to .99 (YW-Combined only). As shown in 6.2, the optimal $\delta$'s are 1, 40, and 5 for YearCounts, WordAffinity, and YW-Combined, respectively. For YW-Combined with $\delta = 5$, the best $\alpha$ value is .95. We hold the $\delta$ values fixed when tuning other parameters (e.g. for span prediction) and obtaining remaining results, but do consider other $\alpha$ values for span prediction.

**MaxChronon vs. MaxGranule.** 6.3 Shows the performance on the Wikipedia development set for each of the models using either MaxChronon or MaxGranule for identifying the anchor (recall that the predicted year is the first year of the anchor). The best anchor extraction method fo each model is bolded. MaxGranule seems to protect YearCounts and YW-Combined (which uses YearCounts) from picking up a chronon that has high probability for the wrong reason—most likely, because of noise from false positives in date extraction. WordAffinity is not susceptible to this, and MaxChronon performs much better.




(a)




(b)

Figure 6.3: Precision, Recall and F-score for span prediction with YW-COMBINED for different values of (a) $\kappa$ with the TRIMMEDSPAN method and (b) $\gamma$ with the VARIANCESPAN method.

### 6.2.2 Parameter tuning: span prediction

We fix chronon size and anchor prediction methods for each model to be their best as determined by year prediction in the previous section. We thus consider six variations of the three models WORDAFFINITY, YEARCOUNTS, and YW-COMBINED with the span prediction methods TRIMMEDSPAN and VARIANCESPAN.

**Parameter setting.** Using binned search over parameter values on the develop-

| Model | Anchor | $\delta$ | $\bar{y}$ | $\tilde{y}$ |
|---|---|---|---|---|
| YearCounts | MaxChronon | 1 | 379 | 23 |
| YearCounts | **MaxGranule** | 1 | 219 | 20 |
| WordAffinity | **MaxChronon** | 40 | 68 | 33 |
| WordAffinity | MaxGranule | 40 | 105 | 34 |
| YW-Combined | MaxChronon | 5 | 159 | 15 |
| YW-Combined | **MaxGranule** | 5 | 38 | 12 |

Table 6.3: Year prediction results: Wikipedia development set.

| Model | Anchor | $\delta$ | $\kappa$ | $\gamma$ |
|---|---|---|---|---|
| YearCounts | MaxGranule | 1 | .05 | 1.5 |
| WordAffinity | MaxChronon | 40 | .05 | 1.5 |
| YW-Combined | MaxGranule | 5 | .40, $\alpha$=.91 | 1.15, $\alpha$=.97 |

Table 6.4: Parameter settings for span prediction methods. The chronon size $\delta$ determined in year prediction experiments is included for completeness.

| Model | Anchor | Spans | $P$ | $R$ | $F$ | $\bar{y}$ | $\tilde{y}$ | $PR_0$ |
|---|---|---|---|---|---|---|---|---|
| YearCounts | MaxGranule | TrimmedSpan | 72.3 | 67.4 | 69.8 | 209 | 8 | 126 |
| YearCounts | MaxGranule | **VarianceSpan** | 72.5 | 72.8 | 72.7 | 209 | 9 | 125 |
| WordAffinity | MaxChronon | TrimmedSpan | 43.8 | 50.8 | 47.1 | 68 | 34 | 157 |
| WordAffinity | MaxChronon | **VarianceSpan** | 40.5 | 70.5 | 51.4 | 71 | 36 | 85 |
| YW-Combined | MaxGranule | **TrimmedSpan** | 81.1 | 78.2 | 79.6 | 82 | 7 | 53 |
| YW-Combined | MaxGranule | VarianceSpan | 73.9 | 78.4 | 76.1 | 85 | 11 | 53 |

Table 6.5: Development set results for span prediction. The parameters *kappa* and *gamma* for each method are set based on 6.4.

| Model | Interval $\delta$ | Anchor | Spans | $P$ | $R$ | $F$ | $\bar{y}$ | $\tilde{y}$ | $PR_0$ |
|---|---|---|---|---|---|---|---|---|---|
| Baseline | 1 | 1964 | $\pm$50 yrs | 38.2 | 54.9 | 45.1 | 113 | 38 | 1154 |
| YearCounts | 1 | MaxGranule | VarianceSpan, $\gamma = 1.5$ | 77.8 | 71.6 | 74.6 | 192 | 8 | 427 |
| WordAffinity | 40 | MaxChronon | VarianceSpan, $\gamma = 1.5$ | 47.9 | 79.7 | 59.9 | 84 | 22 | 292 |
| YW-Combined | 5 | MaxGranule | TrimmedSpan, $\kappa = .40$ | 81.1 | 81.0 | 81.1 | 118 | 6 | 260 |

Table 6.6: Test set results for span prediction. Results are shown for the anchor and span prediction methods that worked best for each model on the development set for a test set of size 4000 documents.

ment set, we tune—with respect to F-score—the $\kappa$ and $\gamma$ parameters of TrimmedSpan and VarianceSpan, respectively, for span prediction. For YW-Combined, we also tune the $\alpha$ parameter at the same time. 6.3 shows the results for (a) *kappa* and (b) *gamma* for YW-Combined using MaxGranule; similar searches were run to obtain values for YearCounts with MaxGranule and WordAffinity with MaxChronon. The final set of parameter values is given in 6.4.

6.5 lists the full results for each configuration. The best span prediction method for each model is bolded. The results make it clear that YearCounts and WordAffinity provide ideal complements to each other. YearCounts is often very close to the true year, but when it is off, it is wildly off (largely due to noise). WordAffinity, instead, is usually not as pinpointed (as evidenced by its much higher median value), but it is never as wildly off (as evidenced by the much better mean value). The combined model effectively combines the strengths of both underlying models, and obtains the best results, regardless of the span prediction method. Finally, note that there is not tremendous variation between the span selection methods for a given model: most of the effectiveness comes from the model's ability to hone in on the generally correct portion of the timeline.

### 6.2.3 Test set results

**Wikipedia life span prediction.** Table 6.6 shows results for each model on the test set (4000 documents), using the chronon sizes, anchor prediction and span prediction (and associated parameters) determined with the development sets. As a baseline, we choose 1964—the year that is the most frequent midpoint in the life spans in the training set—as the anchor, and create a span by adding 50 years to either side of it (1914-2014). This baseline is actually quite strong due to the skew in the dataset, as evidenced by the histogram in 3.1 showing the large number of individuals born in the early 20th century and thus reaching life midpoints in the mid-20th century. All of the models easily beat this baseline, and again we see that YW-Combined performs best of all. It obtains a very low median error: over half of the documents are classified to within six years of their true midpoints. Only 260 out of 4000 documents (6.5%) are totally off the mark.

# Chapter 7

# Discussion

For the temporal prediction task we dicover interesting phenomena such as "tim warps" or "wormholes". This phenomenon has been studied in other areas of text mining such as geolocating documents on the earth's surface. Clements et al. discovered this in Flickr geotags an tried to trace these to different places on earth [35]. We analyse such time warps for the time-stamp prediction task.

## 7.1 Time Warps

Some of the documents for which our model predictions are off by huge margins show interesting wormhole like trends. These are prominent in wiki-year documents due to their terseness as these are list of events that happened in a given year. Besides the models trained on wiki-bio set add to this phenomenon as the context for the two datasets are slightly different. A cluster of dev event years from between 250 to 150 A.D. (e.g. wiki-years 234, 214, 152, 156 etc.) are predicted to be in 2nd century B.C. (200 B.C. to 150 B.C.) by our model. These event years are very short with an average length of 40-50 words per document including. The discriminatory tokens present in these texts are: *Roman, Empire, Kingdom, Han, Dynasty, China, Selucid, Greek, etc.*. In the 200 B.C. to 150 B.C. period all the documents in training set are about Greek/Selucid, Roman and Chinese (mostly from Han dynasty) emperors/personalities (e.g. Attalus I, Eratosthenes, Plautus, Emperor

Gaozu of Han, Emperor Hui of Han, Zhang Qian, Emperor Wen of Han etc.) and contain similar prominent terms as the wiki-year event texts. This common collection of terms leads to the model mapping wiki-year texts to 2nd century B.C. Although these terms are present in the chronons representing 2nd century A.D. too, their proportion in the overall chronon size is very small. The dev document being small assign very high weight in these terms which forces the model to choose chronons which have higher proportion of these words leading to the chronons in 2nd century B.C.

Another interesting prediction is of a cluster of short documents containing similar terms from 200 A.D. to 800 A.D. to mid 6th century A.D. The short wiki-year texts (e.g. wiki-years 246, 822, 486, 750 etc.) contain co-occurring set of terms *Bynzatine, Empire, Roman, Arab, Conquest, Islam, Caliphate*. These short year events text (around 40 to 50 words long on an average) contain events related to mostly Byzantine wars, emperors, Islamic/Arab conquest, Caliphates etc. This is mapped to mid 6th century A.D. period that predominantly contains biographies of Islamic Caliphates (e.g. Abd al-Malik, Abu Bakr, Ali, Umar etc.) and Byzantine emperors and prominent personalities (e.g. Maurice, Fausta, Constans II etc.) which has predominant terms such as: Byzantine, Empire, Caliph, Islam, conquest etc.

## 7.2 Discriminative Words

Table 7.1 shows the top 25 words in the descending order of their strengths. These words are only those that are present in both training and test set. The predictive strength score $S^w_{predictive}$ of a word $w$ is calculated as average prediction error of all the documents that contain the word $w$. The majority of the top 25 predictive words in table 7.1 are

| Most Predictive | Least Predictive |
|---|---|
| meriwether, komatsu, capote, cranmer, payload, morelos, kido, stopes, sap, laila, hem, shakuntala, anthrax, scooby, crayon, plutarch, sampaguita, woodbury, untimely, teleplay, tele, electorates, derivatives, polygram, wavelength | oneself, primari, ssu, thebes, porphyry, lysias, confucius, morality, romana, matteo, unbroken, goodness, timpul, tarii, grout, sinop, cynical, tub, crates, lantern, bite, phila, transaction, corporeal, conciliation |

Table 7.1: Top 50 most predictive and least predicitve words in the descedning order of their stregths on the dev set documents for the wiki-bio dataset.

uncommon nouns. They are mostly not-so-used last names or famous titles e.g. *capote*, *komatsu*, *cranmer* etc. The least predictive ones are common words in majority of the cases e.g. *goodness*, *oneself*, *morality*, *tub*, *crates*, *lantern* etc. The uncommon among the least predictives are mostly ones that are present in just one or two documents for which our model performs very poorly due to getting trapped into one of the time warps present; and it is highly likely that these words might be inducing those warps due to their predominance and "uncommonanlity". Words such as *tele*, *wavelength*, *electorates*, *teleplay*, *sap* (the company) etc. have strong temporal connection as these words have never been used before 19th century and it is words like these that provide predictive strength to our models.

# Chapter 8

# Conclusion

We have shown that it is possible to perform accurate temporal resolution of texts by combining evidence from both explicit temporal expressions and the implicit temporal properties of general words. We create a partitioned timeline with language models in each partition; these are used to determine a probability over the timeline for new test documents. The language models themselves are trained on texts that are labeled with times based on their content rather than their publication date, allowing us to temporally resolve documents in the range 3800 B.C. to 2010 A.D. For the time span task the best model, which combines explicit and implicit indicators, obtains an $f$-score of 81.1% on predicting the life spans of individuals based on their Wikipedia biographies.

The time warps and least and most predicitive word sets provide us insight into the temporal distribution of the training set. As we would expect the least predictive words were the most common words such as *oneself*, *morality*, *confucius* etc. The uncommon words in the least predcitive set essentailly contributed to the time warp phenomena and lead to a very high amount of error. The most predictive words were the uncommon as well as ones having implicit temporal properties associated with them. Words like *tele*, *wavelength*, *electorates*, *teleplay* have strong temporal association and as expected are found among the top 20 most predictive words.

For time-stamp task our best model was able to predict the mid-life span of Wikipedia

biographies upto a mean error of 37.35 years and median error of 22.0 years over the test set, given the biographies spanned from year 3800 B.C. to present day. The best model gave a mean-error of 22.89 years and median error of 19.0 years for predicting the publication dates of English short stories obtained from Guttenberg over test set. The publication date of the stories lied between the years 1798 to 2008. We also predicted the year of occurrence of Wikipedia "year events" pages. We were able to locate the year of occurence with a mean error of 37.9 years and a median error of 20.0 years.

# Chapter 9

# Future Work

There are a number of ways to improve the present approach. In building our time-sensitive language models, we can match between time spans on annotated documents and the chronons on our timeline in different ways. In this work we include in each pseudo-document the full counts of any document whose timespan overlaps it. We can instead give fractional counts proportional to the degree of overlap, akin to our span-based evaluation measures. Inspired by recent work in positional language models [34], we can also posit some form of density function across the years within the annotated time span, or centered on a text with only a single year annotated such as publication date. These modifications are worth trying and are part of out planned future work.

Smoothing of LMs of chronons around their neighborhood might be of help here. It is likely that chronons in proximity to each other will have similar LMs. Taking into account the proximity of queries/ documents while generating their LMs have improved accuracy of models in the past [13, 45]. It remains to be seen whether a similar approach taking proximity into account can imporve the model accuracies. The models can be built using n-grams instead of unigrams. Words such as *New York* have their own temporal signature compared to their unigram counterparts (*New* and *York*).

The temporal models discussed here can be used for word sense dismabiguation. In recent years there have been increased interests in *grounding* words to real world properties.

Words have been traditionally defined in terms of other words, a cyclic (though useful) definition. Recent works have tried to bind words to conceptual representations [4], models of space [43] and time. This is useful for word sense disambiguation. For example, the word *a*pple used before 19th century will mean *the fruit apple* with a very high probability compared to that in the late 19th century. It can mean *Apple Inc.* or the record label *Apple* as well.

An interesting future work can be the exploration of possible space of priors that can be assigned to the chronons in the Bayes model. The prior used here takes into account the number of documents assigned to each chronon. It is worthwhile to looke for alternative prior formulation. An obvious prior that can be explore is the chronon's ML in the training set. A prior that takes into account the degree of overlap of the documents assigned to a chronon is also worth trying.

At present we obtain the smoothing parameters by searching through the possible parameter space. We might be able to avoid this by using Empirical Bayes smoothing techniques [30]. Another possibility is to assume a continuous distribution form (e.g. mixture of gaussians) for each chronon and document over the words similar to that of Khokhar et al. [42]. It would be computationally intensive to model mixture of gaussians for each chronon and document though it would avoid a possible search over parameter space. It would also need large amount of evidence for all the chronons. For the present training set our choron size (in terms of number of tokens) is typically very small in the B.C. years region as shown in figure 3.1. This might lead to problems in fitting a mixture of gaussians over these chronons.

Another opportunity with our approach that is worth exploring is to bootstrap the

training of the model by using the dates obtained via temporal expression taggers (e.g. Heidel Time [44]). This would enable the creation of a much larger and more representative corpus from which to estimate the chronon-specific language models. There are plenty of such texts available, including the many non-biographic articles in Wikipedia and many freely available books written in the 19th and early 20th century that are about much earlier time periods (and which have many explicit temporal expressions). This analysis is part of our planned future work.

Google N-grams dataset can be used to train the present models. This dataset is obtained from scanning around 5 million books and contains books published from 15th century till date [16]. They follow a similar approach of dividing the timeline into single years and assign books to these years according to their publication date. Though it has considerably more documents than our present training set it has books only from 15th century onwards. This makes it not so useful for predicting time stamps in the B.C. region of texts.

Separating out words that have different periods of temporal behavior is a promising future work. For example, *breakfast* and *dinner* cycle every day, *weekend* cycles every week, *paycheck* cycles every month, and *winter* and *summer* cycle every year—all of these are at much finer granularities than the year-by-year models we use. Data is now available for them via constant newswire and Twitter feeds, and it is likely to be quite important to detect and tease apart such periodic properties of words, especially for more fine-grained temporal resolution. This can be used to assign much finer time stamps to text especially micorblogs or tweets. We might be able to tell apart morning form evening tweets or weekday from weekend tweets. People typically have different routines for weekdays and weekends which

leads to the words used over the two periods containing implicit temporal cues.

Finally, the models discussed here can also be used for identifying shifts in users political or social bias over issues with time.

# Bibliography


[1] James F. Allen. Maintaining knowledge about temporal intervals. In *Communications of the ACM, Volume 26 Issue 11*, 1983.

[2] O. Alonso, M. Gertz, and R. Baeza-Yates. Clustering and exploring search results using timeline constructions. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 97–106, 2009.

[3] Geoffrey Andogah. *Geographically Constrained Information Retrieval*. PhD thesis, University of Groningen, Groningen, Netherlands, May 2010.

[4] M. Baroni, B. Murphy, E. Barbu, and M. Poesio. Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science*, 34(2):222–254, 2010.

[5] Christopher M. Bishop. Mixture models and em. *Pattern Recognition and Machine Learning*, pages 423–455, 2006.

[6] David Blei, Chong Wang, and David Heckerman. Continuous time dynamic topic models. In *Intl. Conference on Machine Learning (ICML)*, 2008.

[7] David M. Blei and John D. Lafferty. Dynamic topic models. In *23rd Intl. Conference on Machine Learning (ICML)*, 2006.

[8] Xiaofan Cao. *Model selection based on expected squared Hellinger distance*. PhD thesis, Department Of Statistics, Colorado State University, 2007.

[9] Nathanael Chambers and Dan Jurafsky. Jointly combining implicit constraints improves temporal ordering. In *EMNLP*, 2008.

[10] Yun Chi, Shenghuo Zhu, Xiaodan Song, Junichi Tatemura, and Belle L. Tseng. Structural and temporal analysis of the blogosphere through community factorization. In *Proc. of 13th SIGKDD*, 2007.

[11] W. Dakka, L. Gravano, and P. Ipeirotis. Answering general time-sensitive queries. *Knowledge and Data Engineering, IEEE Transactions on*, (99), 2010. Pre-print.

[12] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[13] F. Diaz. Regularizing query-based retrieval scores. *Information Retrieval*, 10(6):531–562, 2007.

[14] Junyan Ding, Luis Gravano, and Narayanan Shivakumar. Computing geographical scopes of web resources. In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB)*, pages 545–556, 2000.

[15] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287, 2010.

[16] Gray M. and Team B. and Pickett J. and Hoiberg D. and Clancy D. and Norvig P. and Pinker S. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014), 2011.

[17] Ralph Grishman, Silja Huttunen, and Roman Yangarber. Real-time event extraction for infectious disease outbreaks. In *Proceedings of the second international conference on Human Language Technology Research (HLT)*, San Diego, 2002.

[18] L. Hartung Imran Saleemi and Mubarak Shah. Scene understanding by statistical modeling of motion patterns. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, 2010.

[19] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. In *Journal of Documentation*, Cambridge, 1972.

[20] Franciska De Jong, Henning Rode, and Djoerd Hiemstra. Temporal language models for the disclosure of historical text. In *XVIth International Conference of the Association for History and Computing (AHC)*, pages 161–168, 2005.

[21] R E Kalman. A new approach to linear filtering and prediction problems. *Journal Of Basic Engineering*, 82:35–45, 1960.

[22] N. Kanhabua and K. Nørvåg. Improving temporal language models for determining time of non-timestamped documents. In *12th European conf. Research and Advanced Technology for Digital Libraries (ECDL)*, pages 358–370, 2008.

[23] N. Kanhabua and K. Nørvåg. Determining time of queries for re-ranking search results. In *14th European conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 261–272, 2010.

[24] A. Kulkarni, J. Teevan, K.M. Svore, and S.T. Dumais. Understanding temporal query dynamics. In *Proceedings of the 4th ACM International conference on Web Search and Data Mining (WSDM)*, pages 167–176, 2011.

[25] S. Kullback and R.A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[26] Abhimanu Kumar, Matthew Lease, and Jason Baldridge. Supervised language modeling for temporal resolution of texts. In *Proceeding of the 20th ACM Conference on Information and Knowledge Management (CIKM)*, pages 2069–2072, 2011.

[27] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119. ACM, 2001.

[28] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of SIGIR*, pages 111–119, 2001.

[29] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, New Orleans, 2001.

[30] Glen H. Lemon and Richar G. Krutchkoff. An empirical bayes smoothing technique. *Biometrika*, 56:361–365, 1969.

[31] X. Li and W.B. Croft. Time-based language models. In *12th International Conference on Information and Knowledge Management (CIKM)*, pages 469–475, 2003.

[32] X. Liu and W.B. Croft. Cluster-based retrieval using language models. In *27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193, 2004.

[33] Vitor Loureiro, Ivo Anastcio, and Bruno Martins. Learning to resolve geographical and temporal references in text. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*, pages 349–352, 2011.

[34] Y. Lv and C.X. Zhai. Positional language models for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306, 2009.

[35] Arjen P. de Vries Maarten Clements, Pavel Serdyukov and Marcel J. T. Reinders. Finding wormholes with flickr geotags. *Lecture Notes in Computer Science*, 5993/2010:658–661, 2010.

[36] Christopher D. Manning, David Hall, and Daniel Jurafsky. Studying the history of ideas using topic models. In *ACL*, 2008.

[37] Bruno Martins. *Geographically Aware Web Text Mining*. PhD thesis, University of Lisbon, 2009.

[38] Pawel Mazur and Robert Dale. Wikiwars: A new corpus for research on temporal expressions. In *EMNLP*, Massachusetts, 2010.

[39] Perseus-Project. American civil war collection. `http://www.perseus.tufts.edu/hopper/collection?collection=Perseus:colle%ction:cwar`.

[40] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, 1998.

[41] James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, David Day, Lisa Ferro, Robert Gaizauskas, Marcia Lazo, Andrea Setzer, and Beth Sundheim. The TimeBank corpus. *Corpus Linguistics*, pages 647–656, 2003.

[42] Mubarak Shah Salman Khokhar and Imran Saleemi. Similarity invariant classication of events by kl divergence minimization. In *Proceedings of the 13th International Conference on Computer Vision (ICCV)*, Barcelona, 2011.

[43] Mike Speriosu, Travis Brown, Taesun Moon, Jason Baldridge, and Katrin Erk. Connecting language and geography with region-topic models. In *1st Workshop on Computational Models of Spatial Language Interpretation*, 2010.

[44] Jannik Strtgen and Michael Gertz. Semeval '10 proceedings of the 5th international workshop on semantic evaluation. In *SemEval 2010 Proceedings of the 5th International Workshop on Semantic Evaluation*, 2011.

[45] T. Tao, X. Wang, Q. Mei, and C.X. Zhai. Language model information retrieval with document expansion. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pages 407–414, 2006.

[46] UNC. Documenting the american south. `http://docsouth.unc.edu`.

[47] Marc B. Vilain and Bolt Beranek. A system for reasoning about time. In *Proceedings of the Second National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, 1982.

[48] Walt-Whitman. Walt whitman archive. `http://www.whitmanarchive.org/`.

[49] Xuerui Wang and Andrew McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Knowledge Discovery and Data-mining (KDD)*, 2006.

[50] Wikipedia. American civil war texts. `http://en.wikipedia.org/wiki/Portal:American_Civil_War`.

[51] Benjamin Wing and Jason Baldridge. Simple supervised document geolocation with geodesic grids. In *ACL-HLT*, 2011.

[52] Fei Wu, Raphael Hoffmann, and Daniel S. Weld. Augmenting wikipedia-extraction with results from the web. In *AAAI*, Georgia, 2010.

[53] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[54] Jianwen Zhang, Yangqiu Song, and Changshui Zhang. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In *Proceedings of the 16th ACM SIGKDD*, 2010.

[55] Y. Zhao and J. Zobel. Entropy-based authorship search in large document collections. *ECIR*, pages 381–392, 2007.

[56] Y. Zhao, J. Zobel, and P. Vines. Using relative entropy for authorship attribution. *Third Asia Information Retrieval Symposium (AIRS)*, pages 92–105, 2006.