Toward Enhanced Metadata Quality of Large-Scale Digital Libraries: Estimating Volume Time Range

Siyuan Guo, Indiana University
Trevor Edelblute, Indiana University
Bin Dai, Indiana University
Miao Chen, Indiana University
Xiaozhong Liu, Indiana University

Abstract
In large-scale digital libraries, it is not uncommon that some bibliographic fields in metadata records are incomplete or missing. Adding to the incomplete or missing metadata can greatly facilitate users' search and access to digital library resources. Temporal information, such as publication date, is a key descriptor of digital resources. In this study, we investigate text mining methods to automatically resolve missing publication dates for the HathiTrust corpora, a large collection of documents digitized by optical character recognition (OCR). In comparison with previous approaches using only unigrams as features, our experiment results show that methods incorporating higher order n-gram features, e.g., bigrams and trigrams, can more effectively classify a document into discrete temporal intervals or "chronons". Our approach can be generalized to classify volumes within other digital libraries.
Keywords: Temporal classification; Metadata enhancement; HathiTrust Research Center
Citation: Guo, S., Edelblute, T., Dai, B., Chen, M., Liu, X. (2015). Toward Enhanced Metadata Quality of Large-Scale Digital Libraries: Estimating Volume Time Range. In *iConference 2015 Proceedings*.

Acknowledgements: The authors would like to acknowledge the HathiTrust Research Center for providing the dataset and accompanying metadata.
Contact: siyguo@indiana.edu, tedelblu@indiana.edu, bindai@indiana.edu, miaochen@indiana.edu, liu237@indiana.edu

1 Introduction

Metadata is a special type of data that describes data. In the age of Big Data, the role of metadata has become more prominent–it is obvious that big data needs high-quality metadata description as it becomes less and less possible for humans to go over all the data (if human readable) with the exponential growth of data sets. In this study we try to enhance metadata records (publication dates) by developing a temporal classification approach for a large-scale digital library. This approach can help assign temporal information given the full-text content of a library item, such as a book. Temporal classification of text, whether it be webpage content, wikipages, or volumes from Project Gutenberg, has had a growing interest in areas of information retrieval and computational linguistics. The addition of temporal information has been used to significantly improve query search results.

Here we contribute methods that incorporate new, higher order n-gram features, specifically bigrams and trigrams, to successfully predict a given document's membership to a chronon. We were presented with an opportunity to work with the public domain corpora of the HathiTrust (HT) digital library which is the world's largest digital library with scanned volumes from research libraries covering a wide span of time, from pre-1500 to present. The broad body of digital volumes in HT provides an opportunity to develop a temporal classification approach for this large-scale digital library as well as similar digital libraries. For our data set, 13% of publication date is missing from the metadata records. It thus serves as a good corpus for temporal classification algorithm application.

2 Related Work

Adding temporal information to documents can aid user queries with regard to information retrieval (Alonso, Gertz, & Baeza-Yates, 2007; Kanhabua & Nørvåg, 2008). This is potentially highly beneficial to digital libraries. Prior to any temporal classification of text, it is critical to define temporal expressions of time as a continuous variable in nature. In this study, we split the document timeline into 12 chronons, i.e. 12 temporal categories.

Previous studies have employed different types of text characteristics to estimate the time period of the authored text. (Alonso et al., 2007) contribute a time-based document clustering algorithm, *TCluster*, that includes ranking of query terms based on co-occurrence or calculating the distance of the query terms to the temporal entity.

de Jong, Rode, and Hiemstra (2005) discuss the building of temporal profiles of words combined with *temporal language models* to bridge historical search terms and their modern variants thus producing a diachronic lexical database (WordNet). The authors outline a set of requirements needed to classify a document with an unknown date within a corpus of documents from a given time span. The authors provide methods for evaluating the document classification based on two approaches for the dating task–comparison on document level and comparison on temporal partition level.

Kanhabua and Nørvåg (2008) employed semantic-based preprocessing, including part-of-speech tagging, collocation extraction, word sense disambiguation, concept extraction, and word filtering. The authors also used temporal entropy as a term weighting scheme providing an alternative to *IDF* for determining the importance of a word in a given document.

Kumar, Lease, and Baldridge (2011) use the implicit text cues available in a document combined with any explicit date expressions in the text to produce an improved model to more accurately predict the dating of a document. In their follow-up study, they investigate the feasibility of using text alone, i.e. without explicit temporal cues, to assign dates to documents (Kumar, Baldridge, Lease, & Ghosh, 2012). The authors apply a unigram language model to compute the document's similarity to a given chronon. Two metrics (KL-divergence, document-likelihood) and three smoothing techniques (Jelenik-Mercer, Dirichlet, Chronon-specific) are throughoutly investigated. Their results show that Jelenik-Mercer works the best among the three smoothing techniques, and KL-divergence metric works similar to the document-likelihood approach.

Compared to previous studies, our study uses a richer feature set: both temporal cues in text and document-similarity measures based on higher order n-grams. Also, to the best of our knowledge, we have not encountered any study of temporal classification that dealt with corpora as large and noisy as ours.

3 Dataset

3.1 OCR Text Files

We obtained the full set of the non-Google digitized public domain portion of the HathiTrust digital library, which is a collection containing over 255,000 volumes. The text of the volumes are harvested using the HathiTrust Research Center Data API (HathiTrust Research Center, 2014a). The text files are optical-character-recognized files resulting from scanned images of the volumes.

3.2 Metadata

Each volume in HathiTrust digital library is accompanied by a bibliographic metadata record formatted in XML using the MARC (MAchine-Readable Cataloging) standards. The MARC bibliographic metadata describe the individual volumes using datafields that include a unique record identifier, title, and the data-rich, fixed-length element '008'. Data element '008' contains the publication-specific data, i.e. publication date, publication place, material type, and language. We are able to obtain nearly 245,000 MARC XML records from the HTRC Solr API (HathiTrust Research Center, 2014b).

Of all the metadata records, just over 27,000 were found to be either missing an explicit publication year for the volume or populated with an invalid or incomplete date. Using the MARC metadata, six percent or nearly 15,000 volumes were identified as a language other than English. The remaining 203,000 volumes span a publication date range from 1520 to 2002. Figure 1 visualizes the distribution of document counts by year. We identified that nearly 175,000 or 86% of the records indicated a publication date in the last quarter of the 19th century through 1922. The remaining fifteen percent of the dataset is sparsely distributed across the years 1520-1877 and 1923-2002. The sudden cutoff at 1922 results from limited access to volumes after 1923 due to copyright restrictions.

3.3 Chronons (Classification Labels)

Chronons are buckets for splitting publication years into several categories. They are also the class labels for the temporal classification task. We parse the XML metadata files and extract values of the "publish date" field. Similar to previous works, we partition continuous time into discrete units called "chronons".

Chronon	pre-1840	1840-1860	1861-1876	1877-1887	1888-1895	1896-1901
# of Vols	19,779	20,113	22,508	22,169	22,957	22,636
Chronon	**1902-1906**	**1907-1910**	**1911-1914**	**1915-1918**	**1919-1922**	**post-1923**
# of Vols	21,700	19,776	22,169	21,101	21,833	1,192

Table 1: Number of volumes in each chronon.

Previous works mostly used a fixed number of years as a unit where the smallest temporal granularity is usually a single year (Alonso, Gertz, & Baeza-Yates, 2009). However, as shown in Figure 1, the volumes are distributed unevenly over the years. If we were to adopt the granularity of previous works, the distribution of each chronon (an atomic interval) would duplicate Figure 1. To resolve this challenge, we created 12 date range splits, or chronons, that match a more equitable distribution of documents across the corpus. Table 1 shows time range and number of volumes in each chronon. We should point out that the latest chronon (1923-present) only has 1,192 documents due to copyright restrictions.

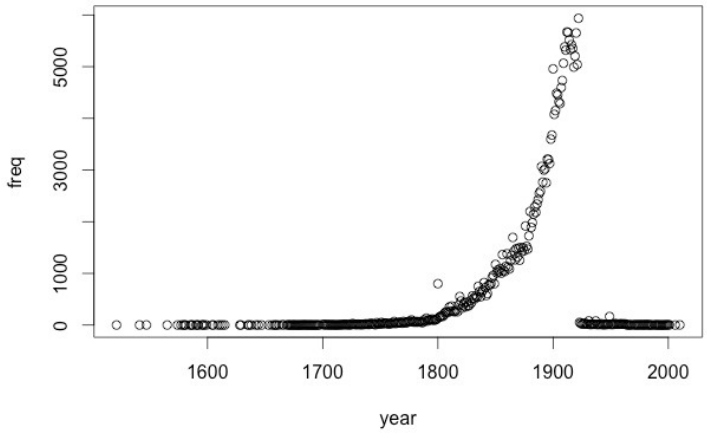


Figure 1: Document counts by year.

4 Feature Extraction

Having identified our twelve dependent variables, we needed to extract some independent variables as features to perform the date prediction. In this early stage, we obtained two types of features from the document: temporal cues and text cues, both of which are extracted from full-text content.

4.1 Temporal Cues

A straightforward approach is to find existing temporal information in a document and use it to predict the publication date. Here we propose two temporal features:

First-Date-In-Text (FD) is the first occurrence of a numeric year or a near-year-like string within each document and its assignment to one of the twelve chronon. We devised this method based on our observation that most documents typically begin with a copyright or title page which may contain the publication date. In order to obtain first-date-in-text, we searched each document to find the first 4-digit number that was in the range 1400 to 2000. From these results, each document was assigned its respective chronon. Based on our initial familiarity with the corpus contents, we wanted to address some of the date-to-string OCR errors we had witnessed. We therefore extracted the first occurrence of all year-like values, i.e. a 4-character string with at least one digit in it, e.g. *l9QO* (el-nine-cue-o). Each instance of any potential OCR error was thus greedily modified using the mapping shown in Table 2.

OCR Errors	l	J	Q	O	o
Replacement	1	1	0	0	0

Table 2: A replacing map for OCR errors of digits.

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	TE	Freq	D
fi	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	634395	5.80
fl	0.53	0.07	0.15	0.07	0.05	0.03	0.07	0.01	0.02	0.00	0.01	0.00	0.37	77652	1.79
ſ	0.45	0.05	0.22	0.07	0.07	0.03	0.05	0.01	0.03	0.00	0.02	0.00	0.31	288633	1.72

Table 3: Distribution of 'fi', 'fl' and 'ſ' across 12 chronons ($c_{1...12}$)

We again checked whether the resulting 4-digit year value was in the range 1400 to 2000 or not. As a result, the near-year-like string "l9QO" would be converted to 1900 and assigned to the chronon label 1896-1901.

Date-Distribution-In-Text (DD) is the distribution of chronons in the document. Excepting some subset of fiction novels, we assumed that most examples of numeric year values in the document would be less than the actual publication year. We believed that date distribution throughout the text may reflect temporal characteristics about the publication date of the document itself. We used the same method as above to extract dates from texts. However, instead of extracting the chronon label of only the first date, we extracted chronon labels of all the dates shown in text. Then we counted the frequency of each chronon label and got a distribution across the 12 chronons.

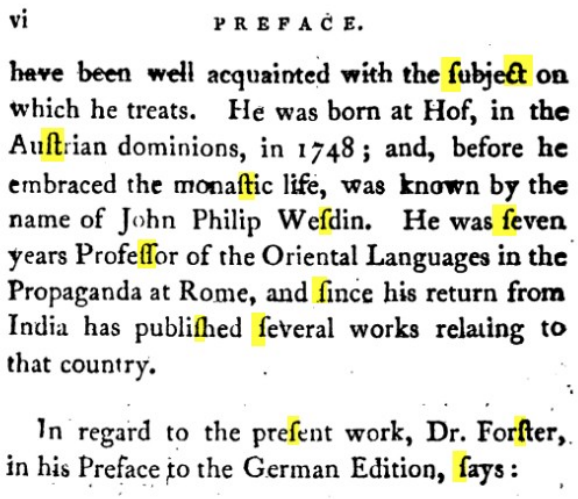
4.2 Text Cues

Text content clues can be used for classification, especially for documents without any copyright page or other temporal information indicating publication date.

Bag of words, unigrams, bigrams, trigrams. Previous works have successfully classified or predicted document publication dates using text cues (de Jong et al., 2005; Kanhabua & Nørvåg, 2008; Kumar et al., 2012). They built a temporal language model based on unigrams to capture topic information of each document and chronon. However, their language model faces the challenge of data sparsity when bigrams and trigrams were included. Though previous works have made exceptional efforts to overcome this challenge by optimizing smoothing techniques, all of them are only using unigrams. In our paper, we tried to include higher order n-grams without being trapped by the sparsity of data. As a compromise, we used a bag-of-words (BoW) model instead of a generative language model. Each document and chronon is represented as bag-of-words (BoW) vector, where each word can be unigram, bigram or trigram, and each cell of the vector is the relative frequency of each word. For smoothing out the zero values, we simply assigned zero-frequency words a default frequency of value (0.0001).

OCR error counts. After inspecting the raw corpora, we found that OCR errors are prevalent across the whole dataset, and interestingly, certain OCR errors are chronon-specific. An example of these clustered, time-based errors are typographical ligatures. There began a rise in the use of ligatures with the advent of movable type (15th C.). Ligature use began to be phased out in the 1800s and became rare by the 1950s with the use of sans-serif type. Some of these might be described as the 'long S', and 'e in o', etc. Figure 2 is a sample from a document in the HathiTrust with the ligatures highlighted. It is worthwhile to quantify OCR errors as a feature. Previous research captured OCR errors using a large amount of replacement and transformation rules and a spellcheck component (Auvil, 2011). However, OCR correction scripts that reference replacement rulesets can create errors by replacing text that is not an error. The long 'S' of the eighteenth-century is an example where making replacements based on rulesets can be problematic, e.g. {'fame':'same', 'fell':'sell', 'fold':'sold'} (Underwood, 2013). In addition, an extensive ruleset potentially runs slowly in a large dataset like ours. We propose a bag-of-character model to approximately quantify OCR errors. For example, Table 3 shows the distribution of 'fl', 'fi' and 'ſ' across 12 chronons.

We preprocessed each document to extract all n-grams (n=1,2,3) that occurred at least 5 times in the corpus and their term frequencies. The term frequencies were then normalized by document length. Due to the large amount of data, we parallelized this process using GNU Parallel (Tange, 2011) with MapReduce


vi PREFACE.

have been well acquainted with the ſubject on which he treats. He was born at Hof, in the Auſtrian dominions, in 1748; and, before he embraced the monaſtic life, was known by the name of John Philip Weſdin. He was ſeven years Profeſſor of the Oriental Languages in the Propaganda at Rome, and ſince his return from India has publiſhed ſeveral works relating to that country.

In regard to the preſent work, Dr. Forſter, in his Preface to the German Edition, ſays:

Figure 2: A sample page from an 18th century document, highlighted 's' characters are always mis-OCRed as 'ſ', 'fl' or 'fi'.

codes.

4.3 Similarity Measures

Based on the BoW models as described previously, we calculated normalized frequencies of every term in every document and chronon ($P(w|d)$ and $P(w|c)$), so that each document and chronon is represented as a vector. For each chronon, text of all volumes in this chronon is concatenated as a big chronon document. We computed the distance/similarity between each document and each of the 12 chronons based on three different metrics: Cosine Similarity (CS), Kullback–Leibler Divergence (KLD) and Normalized Log-Likelihood Ratio (NLLR).

CS. Cosine similarity is most commonly used in measuring the similarity between two documents on the vector space by calculating the cosine of their angles (as shown in Equation 1). Two vectors with the same orientation have a CS score of 1 whereas vectors sharing no common dimensions have CS score of 0. The CS score between a document d_i and a chronon c_j is:

$$CS(d_i, c_j) = \frac{\sum_{w \in d_i} P(w|d_i) \cdot P(w|c_j)}{\sqrt{\sum_{w \in d_i} P(w|d_i)^2} \sqrt{\sum_{w \in c_j} P(w|c_j)^2}} \quad (1)$$

where $P(w|d_i)$ and ... are the probability of word w occurring in d_i and c_j respectively. Here we simply make the normalized term frequency as $P(w|d_i)$ such that it can be input to CS, KLD, and NLLR.

KLD. Kullback-Leibler divergence (KL-divergence) is used to measure the distance from a 'true' probability distribution p (BoW representation of a document), to a 'target' probability distribution q (BoW representation of a chronon). Due to computing efficiency, we limited the scope of w to words with non-zero relative frequencies in each document. The KLD similarity between document d_i and chronon c_j is:

$$KLD(d_i \not\| c_j) = \sum_{w \in d_i} P(w|d_i) \cdot \log \frac{P(w|d_i)}{P(w|c_j)} \quad (2)$$

where the notations remain the same as above.

NLLR. Normalized Log-Likelihood Ratio, a normalized variant of KL divergence, was proposed by previous works as a metric for computing similarity between two temporal language models (Kraaij, 2004; de Jong et al., 2005; Kanhabua & Nørvåg, 2008). NLLR similarity between document d_i and chronon c_j is defined as:

Character	Unigram	Bigram	Trigram
ſ	fuch	sessional papers	speaker put the
ä	cfr	the moft	whether the house
ü	thofe	speaker put	house would agree
ö	subpart	the fame	would agree to
ã	thefe	last paid	as amended at

Table 4: Five most distinctive terms.

Character	Unigram	Bigram	Trigram
-	conform	is required	less than the
9	proposal	protect the	that has been
2	compelling	except that	which do not
z	endorsed	subject to	be removed from
5	separately	must also	people in the

Table 5: Five least distinctive terms.

$$NLLR(d_i, c_j) = \sum_{w \in d_i} P(w|d_i) \cdot \log \frac{P(w|c_j)}{P(w|C)} \quad (3)$$

The only difference between NLLR and KLD is that $\frac{P(w|d_i)}{P(w|c_j)}$ in KLD is normalized as $\frac{P(w|c_j)}{P(w|C)}$, where C refers to the whole corpus, and $P(w|C)$ is the normalized frequency of word w within the whole corpus.

Temporal Entropy. Not every word is temporally distinctive. Some words occur equiprobably across the 12 chronons while some other words occur only in certain chronons. To quantify the temporal distinctiveness of a word w_i across chronons in the corpus, we used the notion of temporal entropy (TE) in (Kanhabua & Nørvåg, 2008; Lochbaum & Streeter, 1989) which is formalized as below.

$$TE(w_i) = 1 + \frac{1}{\log N^C} \sum_{c \in C} P(c|w_i) \cdot \log P(c|w_i) \quad (4)$$

where $P(c|w_i)$ is $\frac{count(w_i, c_j)}{\sum_{k=1}^{N_C} count(w_i, c_k)}$, $count(w_i, c_j)$ is the frequency of word w_i in chronon c_j, and $\sum_{k=1}^{N_C} count(w_i, c_k)$ is frequency of word w_i in the whole corpora. So a term that occurs equiprobably across 12 chronons will get the lowest possible TE score: 0, while a term occurring only in one specific chronon will get the highest possible value: 1 (e.g. 'fi' in Table 3).

To a certain extent, the logic behind temporal entropy is very similar to that behind inverse document frequency (IDF) if we consider a chronon as a "document". However, temporal entropy captures extra distributional information other than simply counting how many chronons in which a term occurs. For example, imagine that there are two terms: term A occurring in every chronon once and term B occurring in every chronon once except in pre-1839 where it occurs one thousand times. IDF will give A and B a same score while TE will give term B a much higher score than term A.

Table 4 and Table 5 show the five most/least temporally distinctive characters, unigrams, bigrams, and trigrams. Distinctiveness of word is quantified as the product of its log frequency and its temporal entropy. As shown in the tables, terms with high temporal entropies are chronon-representative OCR errors or literal expressions, while terms with low temporal entropies are common English characters, words and phrases. The most common function words like "to be" verbs and pronouns do not occupy the lowest score positions because their raw frequencies are extremely high even though they distribute uniformly across the 12 chronons.

Temporal entropy was used as a weight-adjusting option to the term frequency which are to be used for similarity measurement. After applying the weight adjustment, the equations for computing document-chronon distance were modified as follows:

$$wCS(d_i, c_j) = \frac{\sum_{w \in d_i} \hat{P}(w|d_i) \cdot \hat{P}(w|c_j)}{\sqrt{\sum_{w \in d_i} \hat{P}(w|d_i)^2} \cdot \sqrt{\sum_{w \in c_j} \hat{P}(w|c_j)^2}} \quad (5)$$

where $\hat{P}(w|d) = TE(w) \cdot p(w|d)$

$$wKLD(d_i /\!\!/ c_j) = \sum_{w \in d_i} TE(w) \cdot P(w|d_i) \cdot \log \frac{P(w|d_i)}{P(w|c_j)} \quad (6)$$

$$wNLLR(d_i, c_j) = \sum_{w \in d_i} TE(w) \cdot P(w|d_i) \cdot \log \frac{P(w|c_j)}{P(w|C)} \quad (7)$$

As shown in Equation 5, each cell of a vector is weighted by its corresponding temporal entropy before computing cosine similarities. For KLD (Equation 6) and NLLR (Equation 7), we use the same weighting approach as (Kanhabua & Nørvåg, 2008).

5 Experiments

5.1 Data Preparation

On each run of our experiments, a 10-fold cross-validation with stratification on chronons was conducted in order to split the whole dataset into training and testing sets. Each document was then converted into a feature matrix using the previously described methods. In total, we have prepared 26 feature sets as input to our classifiers:

- 2 temporal feature sets: FD and DD;
- 24 text feature sets (for document-chronon similarity): Cartesian product of
  - 4 term types: character, unigram, bigram, trigram;
  - 2 weight adjusting options: TE-weighted, not adjusted.
  - 3 metrics: CS, KLD, NLLR;

FD, a multi-class categorical variable, was 'dummified' into 12 boolean variables before being fed into our classifiers. The remaining 25 feature sets, of which each contain 12 numeric variables, were directly used in our classifiers without any normalization because our distance metric already bound them in a certain numerical range.

All data manipulation was performed using a Python library called *Pandas* (McKinney, 2010), and all central store and inter-program data communication was performed via *MongoDB* (with document-IDs as index) (MongoDB, 2010).

5.2 Classifiers

Four classifiers were used for evaluation including a baseline model. The Baseline classifier (BL) simply chose the chronon label of first-date-in-text as prediction. Three other classifiers were logistic regression (LR), decision tree (DT) and support vector machine (SVM) which are all commonly used in classification tasks. For LR, we used L2 regularized logistic regression with one-vs-all scheme. For SVM, we used linear kernel SVM with one-vs-the-rest scheme. Both LR and SVM are from *LIBLINEAR* library, an open-source library to assist with large-scale classification problems (Fan, Chang, Hsieh, Wang, & Lin, 2008). For DT, we used the decision tree classifier from Scikit-learn library (Pedregosa et al., 2011) with default settings.

5.3 Experiment Design

We designed five experiments to answer 5 research questions of effectiveness of different combinations of classifiers, distance metrics and features.

	BL	LR	DT	SVM
F-score	74.8	78.2	78.7	77.0
Precision	79.1	79.9	78.8	79.3
Recall	73.2	77.7	78.7	76.4

Table 6: Mean performance of 4 classifiers using only temporal features.

5.3.1 Using Temporal Cues Alone

Exp 1: Is DD a useful feature? In our first experiment, we only used temporal features, that is, only FD and DD were used. We sought to see how the baseline model performed, and the effect of FD since the baseline model used it as the only feature. The other three classifiers used both FD and DD features, so we were able to see whether the DD feature was useful by comparing those three models against the baseline model.

5.3.2 Using Text Cues Alone

Exp 2: Which text feature works the best? Four levels of N-grams have been used to generate text features. We will compare classifiers using each of four text features.

Exp 3: Is character level N-gram a useful feature? Character level N-gram is proposed mainly to capture OCR errors. In this experiment, we examined the effectiveness of features based on bag-of-characters model by comparing the performance of a classifier that includes character-level text features with a model that does not.

Exp 4: Overall performance with incremental text features. It's also interesting to see how models perform without any temporal cues. Because BL relies on temporal features, we exclude it here from analysis. Initially, we wanted to compare different metrics and different classifiers to find out the most performant classifier-metric combination. Thereafter, we used this combination to investigate the effect of incremental features, in other words, whether the performance would get better after adding more text features such as bigrams and trigrams.

5.3.3 Using Temporal & Text Cues

Exp 5: Overall performance with all features included. In our final experiment, we used both temporal and text features in our analysis. We wanted to test whether two types of features can mutually reinforce the performance of each other.

6 Results

6.1 Exp 1

Table 6 shows the result for Experiment 1. Baseline shows high performance since a majority of documents have copyright or preface pages. Performance of LR, DT and SVM are all significantly better (majorly in recall part) than that of the baseline model. Also, because the baseline model only uses FD feature while the other three classifiers use both FD and DD features, our result indicates that DD is effective in reducing false negatives and boosting classification performance.

6.2 Exp 2

Figure 3 shows that higher order N-grams like bigrams and trigrams are more effective than unigrams in temporal classification (around 20% boost in KLD and NLLR metrics). However, the performance difference between bigram and trigram is marginal or non-significant, while character-level ngrams perform consistently poorly across different classifier-metric combinations.

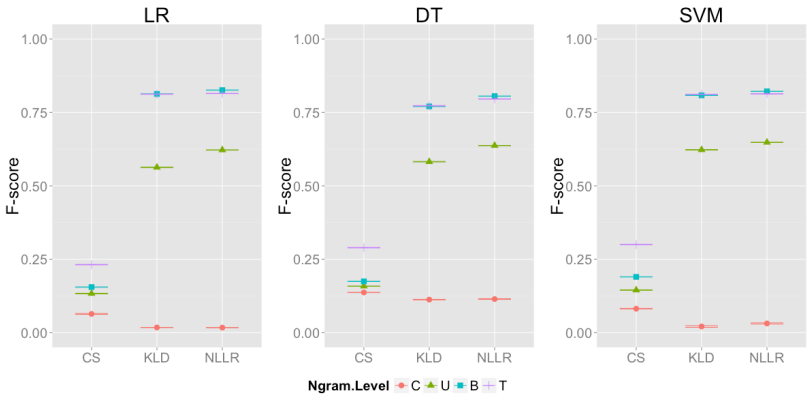


Figure 3: Mean F-scores of 3 classifiers × 3 weighted metrics using only each level of text features.

6.3 Exp 3

After comparing every classifier-metric combination, we found no significant performance improvement for models that include character-level text features. A possible reason is that OCR errors captured by character-level N-grams are already captured by unigrams and bigrams, e.g. Table 4 shows that unigrams and bigrams also capture OCR errors. No extra information is provided when character-level N-grams are included.

6.4 Exp 4

The upper four graphs in Figure 4 depict the result of Experiment 4. The first three graphs are used to compare the performances of different combinations of classifiers and metrics, when using all text features. It shows that KLD and NLLR works consistently better than CS. We then chose the most effective classifier-metric combination (NLLR-LR) to study the effect of incremental text cues. The right-most graph shows that the performance will be significantly improved when a new N-gram feature is added. It indicates that higher order N-gram features like bigram and trigram are indeed useful in temporal classification, and can even outperform temporal features. We also find performances of weighted metrics are significantly better than unweighted ones in most cases.

6.5 Exp 5

The lower four graphs in Figure 4 depict the result of Experiment 5. Similar to Exp 4, the first three graphs are also used to compare the performances of different combinations of classifiers and metrics. In this section, we used all features together (all text features and all temporal features). The performance when using all features in every combination is better than the performance when only using all text features alone. The most effective classifier-metric combination is still NLLR-LR combination. The right-most graph, just like the graph above it, also indicates that the performance becomes better after more features are added. The performance of weighted metrics are also significantly better than unweighted ones. Finally, we can make a conclusion that temporal features and text features can mutually promote the performance of each other.

7 Discussion

7.1 Conclusion

Our paper proposed a high performance system for classifying large volume of OCRed documents into discrete time periods, which can be potentially used for enhancing metadata quality. Our results also support previous research that text cues (even without explicit temporal cues) are useful for temporal classification of documents. Higher order N-grams like bigrams and trigrams are more effective than unigrams. We proposed a new approach, i.e. bag-of-characters, to capture OCR errors as a feature, though character-level N-grams contribute little to classification performance. However, it cannot rule out the role of OCR errors which are

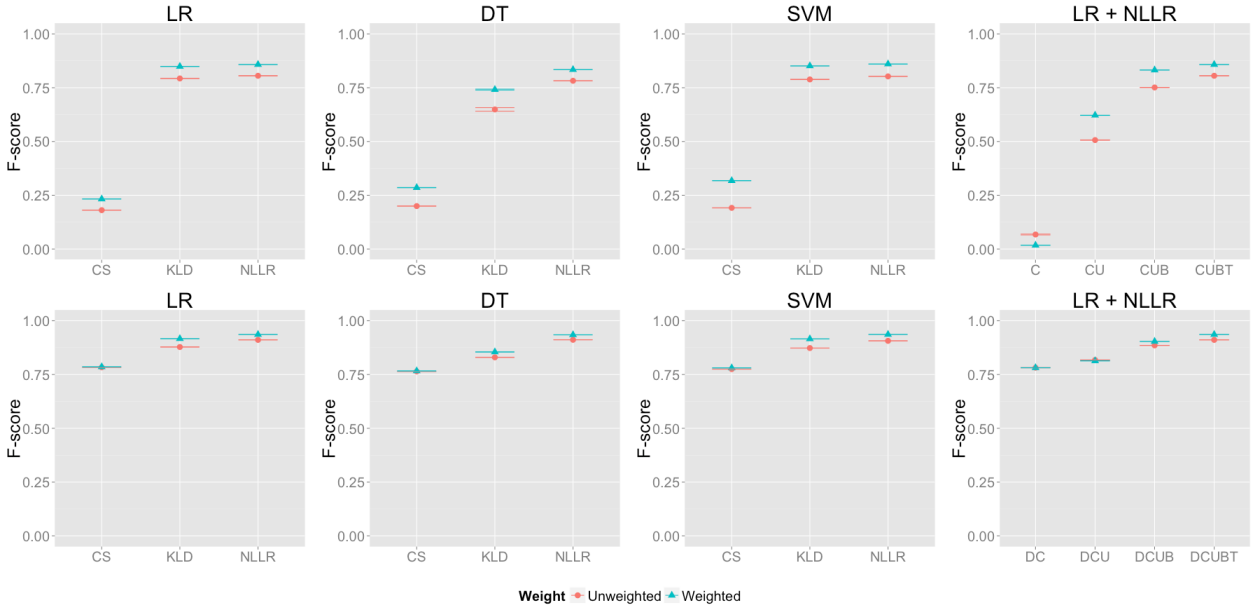


Figure 4: The upper 4 graphs show the result of Experiment 4, and the lower 4 graphs show the result for Experiment 5. The left-most 6 graphs show the performances of different combinations of classifiers and metrics. The 2 graphs in right-most column show the effect of incremental features using the most performant classifier-metric combination (LR+NLLR). Each metric is optionally weighted by temporal entropy. Non-overlapping error bars indicate statistically significant differences between F-score means.

also captured by higher order N-grams. Our research also supports temporal entropy as an effective term weighting strategy.

We have demonstrated this methodology performs reasonably well on HathiTrust digital library data. The broad time range of this data set ensures we have representative features from different chronons. The approach can be generalized to classify time range of documents when applied to other digital library data sets. Moreover, drawing on the trained model from HathiTrust data, people can also classify time range given any documents. This will be an interesting study to conduct in the next step to see whether the performance on external data sets is comparable to HT volumes.

For future work, first, we would like to explore the reason of the low performance of cosine similarity metric by running more experiments. Second, we would try a number of new ways of locating dates in text. The current approach extracts year-like, 4-digit number in text (with minor adjustments for resistance to OCR errors) while strings like '18c', 'MDCCCXXIII' were neglected. Third, we would consider more features from external metadata sources like Wikipedia and Library of Congress. For example, we could include document genre, author's date of birth and author's date of death as features. Fourth, we intentionally used the most simplistic smoothing technique in our paper to better reveal the effectiveness of bigram & trigram features, however, in real application we plan to use Jelinek-Mercer interpolation which is proved by previous works as the most effective smoothing techniques for temporal language models (Kumar et al., 2012). Lastly, classifiers used in our work treat each chronon label as independent multiclass outcomes, however this may not be the case (e.g. chronon 1911 – 1914 may be somewhat dependent on chronon 1907 – 1910). So it is worthwhile to try other classifiers like ordinary regression or C&RT (Classification & Regression Tree) which can capture dependency among outcomes.

References

Alonso, O., Gertz, M., & Baeza-Yates, R. (2007, December). On the value of temporal information in information retrieval. *SIGIR Forum*, *41*(2), 35–41. doi: 10.1145/1328964.1328968

Alonso, O., Gertz, M., & Baeza-Yates, R. (2009). Clustering and exploring search results using timeline constructions. In *Proceedings of the 18th acm conference on information and knowledge management* (pp. 97–106).

Auvil, L. (2011, October). Postprocessing via meandre. In *Ocr workshop presented at OCR Summit Meeting.* Texas A & M University, College Station, TX.

de Jong, F., Rode, H., & Hiemstra, D. (2005). Temporal language models for the disclosure of historical text..

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, *9*, 1871–1874.

HathiTrust Research Center. (2014a). *HTRC data api.* Retrieved 2014-7-8, from http://www.hathitrust.org/htrc/api-guide

HathiTrust Research Center. (2014b). *HTRC solr api.* Retrieved 2014-7-8, from http://www.hathitrust.org/htrc/solr-api

Kanhabua, N., & Nørvåg, K. (2008). Improving temporal language models for determining time of non-timestamped documents. In *Research and advanced technology for digital libraries* (pp. 358–370). Springer.

Kraaij, W. (2004). Variations on language modeling for information retrieval.

Kumar, A., Baldridge, J., Lease, M., & Ghosh, J. (2012). Dating texts without explicit temporal cues. *arXiv preprint arXiv:1211.2290*.

Kumar, A., Lease, M., & Baldridge, J. (2011). Supervised language modeling for temporal resolution of texts. In *Proceedings of the 20th acm international conference on information and knowledge management* (pp. 2069–2072).

Lochbaum, K. E., & Streeter, L. A. (1989). Comparing and combining the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval. *Information Processing & Management*, *25*(6), 665–676.

McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 51 – 56).

MongoDB. (2010). *Mongodb.* Retrieved from http://www.mongodb.org

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Tange, O. (2011, Feb). Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, *36*(1), 42-47. Retrieved from http://www.gnu.org/s/parallel

Underwood, T. (2013). *A half-decent ocr normalizer for english texts after 1700.* Retrieved from http://tedunderwood.com

Table of Figures

Table of Tables