Sub-document Timestamping: A study on the Content Creation Dynamics of Web Documents

Yue Zhao and Claudia Hauff

Delft University of Technology, The Netherlands
{y.zhao-1,c.hauff@tudelft.nl}

Abstract. The creation time of documents is an important kind of information in temporal information retrieval, especially for document clustering, timeline construction and search engine improvements. Considering the manner in which content on the Web is created, updated & deleted, the common assumption that each document has only one creation time is not suitable for Web documents. In this paper, we investigate to what extent this assumption is wrong. We introduce two methods to timestamp individual parts (sub-documents) of Web documents and analyze in detail the creation & update dynamics of three classes of Web documents.

Keywords: Timestamping; Sub-documents; Internet Archive

1 Introduction

Document timestamping is an important step in temporal information retrieval (T-IR) which determines the creation time of documents [4]. Depending on the type of document the creation time can either be extracted directly (e.g. news articles commonly list their creation date) or has to be inferred (as is the case for most Web pages). Such temporal knowledge is essential for a variety of tasks, including document clustering [9, 16, 17, 5], timeline creation [22, 10], and search engine adaptations for temporal queries [19, 14].

Previous studies [9, 15, 16, 17, 5, 11] on document timestamping usually employ a simplifying assumption that each document only has a single creation time. This assumption is suitable for historical documents and news documents, whose content is published at one point in time and is rarely or never updated. Web documents, however, are dynamic; content is added, removed and changed over time. Previous work [12] also shows that users prefer to know the creation time of contents rather than the evolution of Web pages. Therefore, we focus on inferring the creation time of content on Web pages in this paper.

We previously showed that a considerable fraction of Web documents (66.5% of the explored sample) does indeed contain content created at two or more different points in time [23]. Importantly, the creation times of documents' so-called *sub-documents* (a sub-document can be a paragraph or a sentence) can vary widely - content is not created within days, the median time between the oldest and most recent sub-document for the investigated sample of Web documents was 782.5 days. These findings though were derived from a very small set of high-quality Web documents (∼7000) only. Here, we take this work as a starting point and investigate sub-document timestamping on a much larger sample of Web documents (nearly half a million).

Following [23, 15, 16], crawl data from the Internet Archive[1] (IA) is leveraged to obtain ground-truth sub-document creation times. The IA has been archiving Web documents since 1996, and covers a significant but limited part of the Web.

Analyzing the content creation dynamics of Web documents though can only be the first step. Our ultimate goal is to reliably timestamp the sub-documents of all Web documents - independent of their availability in IA. Such fine-grained timestamping would enable large-scale investigations of information diffusion on the Web (e.g. how rumors or specific content spreads) as well as an in-depth exploration of temporal effects on retrieval models (studies of which have so far been restricted to small news corpora). To make this vision a reality, we develop a 2-stage machine learning approach which is not only based on features extracted from individual sub-documents (as done in [23]), but also leverages the relations among the sub-documents in the same Web document. We make the following contributions in this paper:

1. We explore the content creation dynamics of nearly half a million Web documents of varying quality.
2. We gain novel insights into the document factors that play a role in content creation over time.
3. We develop a two-stage machine learning approach to sub-document timestamping, significantly improving upon our previous work [23].

2 Related Work

Due to the importance of document creation times in T-IR, a number of studies have investigated creation time inference (based on the already outlined one-creation-time-per-document assumption). De Jong et al. [9] rely on temporal language models, built from news articles in different time ranges, to determine the most likely creation time range for non-timestamped documents. This approach was extended by Kanhabua et al. [15, 16] and Kumar et al. [17] who introduce additional features, such as temporal entropy and the KL divergence between language models, to improve the accuracy of the temporal language models. Chambers et al. [5] and Ge et al. [11] take temporal expressions appearing in documents and knowledge about the relationships between news documents into account to improve inference accuracy.

The intuition of sub-document timestamping is based on research in Web dynamics which has largely focused on Web evolution [20, 2, 1] and Web crawling [6]. Research on the timestamping of Web documents that takes the Web dynamics into account is largely missing. Jatowt et al. [13] proposed a pipeline for

[1] https://archive.org/

timestamping content based IA data; they however neither analyzed the content changes nor built inference tools to timestamp non-archived Web documents. In our previous work [23], we use a similar pipeline to [13] for IA-based sub-document timestamping. We made a first attempt at analyzing content dynamics (on a few thousand Web documents) and at inference. We now significantly extend our previous work, by analyzing a much larger set of Web documents with varying characteristics and leverage a new machine learning approach to improve the inference of sub-document timestamps.

3 Approach

We now introduce the timestamping pipeline to gain ground truth data and the machine learning approach to infer sub-document timestamps.

3.1 Timestamping Pipeline

Analogously to [23], our timestamping pipeline consists of 4 steps: **(S1)** historical versions extraction, **(S2)** sub-documents extraction, **(S3)** sub-documents timestamping and **(S4)** model training.

Let $D = \{d_1, d_2, ..d_n\}$ be the set of all Web documents d_i we *aim* to collect ground truth data for. For **(S1)** we first extract all historical versions $Hist$ of $d_i \in D$ based on their URLs from the IA. By our definition, two historical versions of a document d_i have to be different from each other, thus we skip archived versions without any content changes. Additionally, since not all Web documents may be available in the IA, we only process those Web documents with records in the IA ($D^{archived}$) in the next steps.

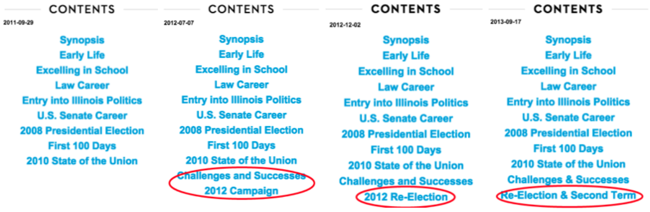
For **(S2)**, all sub-documents are identified and extracted from each d_i in $D^{archived}$ and its corresponding historical versions $Hist_{d_i} = \{d_i^{h_1}, d_i^{h_2}, ..., d_i^{h_m}\}$ where $d_i^{h_1}$ is the earliest version of d_i on IA, and $d_i^{h_m}$ is the most recent version. Every Web document (original and historical versions alike) is then split into sub-documents based on its HTML markup: sub-documents are delimited by `<p>` and `<div>` tags. Only sub-documents with 50+ non-markup characters are considered, to ensure that each sub-document has sufficient content to be correctly matched in the next step.

For **(S3)**, we compare the sub-documents of d_i with all $d_i^{h_i} \in Hist_{d_i}$. The comparison starts at the earliest version $d_i^{h_1}$ and continues in temporal order. We rely on approximate string matching [7] to detect the earliest appearance of each sub-document in d_i within $Hist_{d_i}$.

Finally, in **(S4)** we train our models to automatically estimate the timestamp of unlabelled sub-documents. For experimental purposes we split our datasets into three parts, train on two parts and test the accuracy of the models on the third part.

3.2 Features & Models

We first derive the same 21 features from *each* sub-document that were reported in [23]. These features fall into two categories: (i) term statistics of the sub-document and its sentences (e.g. position and length of the sub-document and its sentences), and, (ii) the values & locations of temporal expressions within the sub-document (e.g. the value of the temporal expression which is earliest or latest). Beyond that, we include two additional types of features: (iii) one numeric feature per year (for the years 1996 to 2012[2]) that expresses the number of temporal expressions containing the respective year in the sub-document, and, (iv) five numeric tense features that express the number of verbs in the respective tense appearing in the sub-document. Thus, in total we compute 44 features per sub-document. All temporal and part-of-speech based features were extracted using Stanford's coreNLP package[3].



CONTENTS
2011-09-29
Synopsis
Early Life
Excelling in School
Law Career
Entry into Illinois Politics
U.S. Senate Career
2008 Presidential Election
First 100 Days
2010 State of the Union

CONTENTS
2012-07-07
Synopsis
Early Life
Excelling in School
Law Career
Entry into Illinois Politics
U.S. Senate Career
2008 Presidential Election
First 100 Days
2010 State of the Union
Challenges and Successes
2012 Campaign

CONTENTS
2012-12-02
Synopsis
Early Life
Excelling in School
Law Career
Entry into Illinois Politics
U.S. Senate Career
2008 Presidential Election
First 100 Days
2010 State of the Union
Challenges and Successes
2012 Re-Election

CONTENTS
2013-09-17
Synopsis
Early Life
Excelling in School
Law Career
Entry into Illinois Politics
U.S. Senate Career
2008 Presidential Election
First 100 Days
2010 State of the Union
Challenges & Successes
Re-Election & Second Term

Fig. 1. Internet Archive based page updates for Barack Obama's biography page.

In [23], the initial 21 features were used in an ensemble learning setup (Random forests), that *classifies* sub-documents into different temporal categories. In this setup, the relations among the sub-documents (and their respective features) are ignored: for each sub-document the timestamp is estimated independently. Intuitively it makes sense to also consider the relations among the sub-documents as some may contain useful temporal features that could also benefit the inference of sub-documents' timestamps appearing in their neighbourhood. As a motivating example for the benefit of sub-document relations, consider Figure 1, which shows how the section headings developed over time for the biography page of *Barack Obama*[4], based on the historical versions extracted from the IA. Generally, content is added or updated towards the end of the document and sections appearing in close spatial proximity are more likely to cover similar time periods compared to sections appearing far apart.

Based on this intuition, we propose a 2-stage model that incorporates the relations among sub-documents as shown in Figure 2. In the first stage we also employ ensemble learning. In the second stage we leverage the predictions of the first stage and input those of spatially neighbouring sub-documents into a Conditional Random Field[5] [18] (CRF), a type of probabilistic graphical model widely used for sequential data labelling. The spatial neighbourhood of a sub-

[2] This time range was chosen due to our experimental data, cf. Section 4.

[3] http://nlp.stanford.edu/software/corenlp.shtml

[4] http://www.biography.com/people/barack-obama-12782369

[5] CRF++: https://taku910.github.io/crfpp/

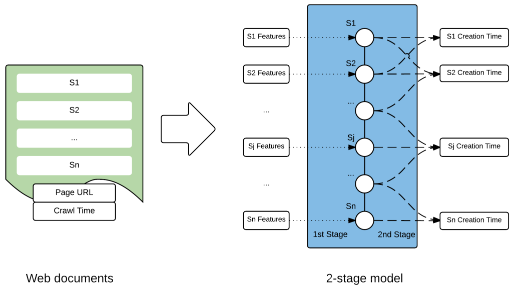


Fig. 2. Overview of our 2-stage model.

document is defined by its position in the HTML markup, instead of the rendered arrangement, to enable efficient processing.

4 Experiments

In our experiments, we leverage a subset of the publicly available ClueWeb12 corpus[6], a Web crawl of more than 700 million pages in early 2012. We investigate three types of Web documents:

`Quality`**:** This set includes all Web documents judged relevant to at least one of the 200 TREC topics released for ClueWeb12 that also appear in IA — 7, 118 documents. This document set was employed in [23]. We consider the documents to be of high quality, as manual judges determined their usefulness to information needs (excluding spam and non-informative pages).

`General`**:** To counterbalance the `Quality` set, we randomly sampled documents from ClueWeb12[7], determined their existence in IA and crawled all historical versions available. Due to IA bandwidth limitations, we continued this process for six weeks, after which we had collected 433,082 ClueWeb12 documents with nearly 3 million (2,961,005) historic versions overall.

`Seen`**:** Lastly, we also sampled a set of "seen" (popular) Web documents, that is Web documents, that were of interest to at least some real users. Here, we were able to exploit the ClueWeb12 crawling strategy: added to the crawl frontier were not only URLs discovered during the standard crawling process, but also URLs that were mentioned in the public Twitter stream during the crawling period. These documents are marked as crawled from Twitter in the ClueWeb12 crawl and we sampled 23,077 of them that were also available in the IA (with 368,106 historic versions).

4.1 Exploratory Analysis

Do the crawl frequencies of documents differ in the IA? Efficient Web crawlers crawl some Web pages (or domains) more often than others, to avoid

[6] http://www.lemurproject.org/clueweb12.php/

[7] Specifically, we sampled from Disk1 of the ClueWeb12 corpus.

re-crawling never changing documents and retaining up-to-date content for regularly changing documents. The IA crawler is no exception. In Figure 3 (right) we plot for the three sets of documents the *average timespan* (in days) between subsequent IA crawled versions[8]. To make the comparison fair (and to remove the IA's changing technological abilities over the years from the comparison), we only consider documents whose first version appeared no earlier than January 2011 in IA and that were crawled at least three times. The results show that our set of `Seen` documents are crawled most frequently, while the set of `General` documents have the largest timespan between subsequent crawls.

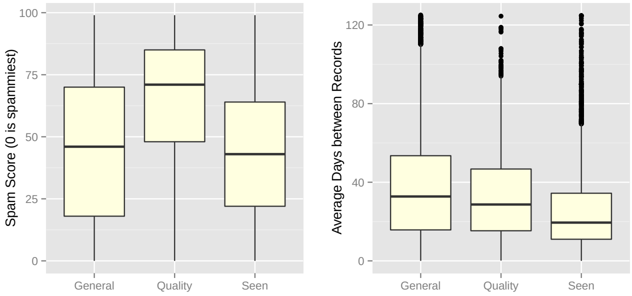


Fig. 3. On the left, the spam score distribution for `General`, `Quality` and `Seen` is shown. On the right, the IA crawling frequency is shown.

To what extent do the document qualities vary across the three sets? Document quality can be measured in many ways, including readability, recency and trustworthiness. We take a practical view on quality and determine the amount of spam each document set includes. We rely on the pre-computed Web spam scores[9] released for ClueWeb12, and plot in Figure 3 (left) the distribution of spam scores. Each document is assigned a spam score, and those scores vary between 0 (most likely to be spam) and 100 (least likely to be spam) — in practice, often documents with a score below 70 are considered to have at least some spam in them. Not unexpectedly, the `Quality` set is mostly spam-free, while the `Seen` documents and `General` documents have a similar amount of spam — indicating that through the public Twitter stream a significant amount of spam entered the dataset. A note of caution though: since the spam scores were derived automatically [8], in future work we will conduct a more qualitative analysis to further verify these findings.

The analyses that follow now are inspired by the questions raised in [23]. Recall though, that only the `Quality` set of documents was investigated before — we experiment with a much larger and more diverse set of data.

What proportion of Web documents is created at multiple points in time? Figure 4 shows the percentage of documents in each set that contains

[8] We mean here *all* versions available on IA, not just those with changed content.

[9] http://www.mansci.uwaterloo.ca/~msmucker/cw12spam/

content (i.e. sub-documents) created at $\{1, 2, .., 10\}$ different points in time. More than 95% of all documents have less than 10 unique creation times. We also observe a marked difference between `Quality` and the other two document sets: less than 35% of `Quality` documents have a single creation time, while this is the case for between 45-55% of documents in `General` and `Seen`. For `Seen` this difference can be attributed to an artifact in the data collection: 38% of `Seen` documents were crawled by the ClueWeb12 crawler *before* they were first archived by the IA (a natural explanation being that people regularly tweet about newly created content on the Web). In these instance we assign the ClueWeb12 crawl time as their creation time. The same though cannot be said about `Quality` or `General` where this is the case in less than 7% of all documents.

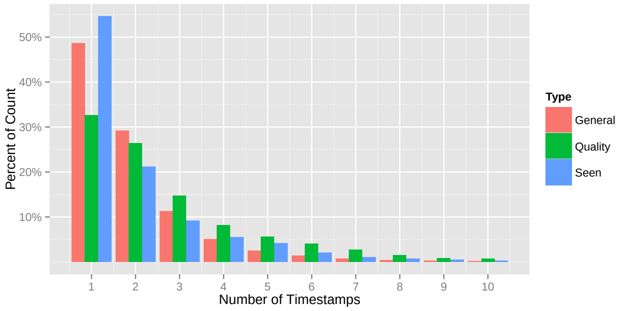


Fig. 4. Number of documents containing content created at different points in time.

How much time passes between content updates? Having determined that there are indeed sufficient Web documents with multiple creation timestamps, we are now concerned with the time that has passed between the first and last creation timestamp of sub-documents in the same document. If changes were mostly made within a few days of the original creation of a document, there would be little need for sub-document timestamping in T-IR applications. In Figure 5 we show for all documents (and all three document sets) with more than 1 creation timestamp how large the timespan between the earliest and latest sub-document creation time is. On average, we observe surprisingly large timespans: 350 days (`Seen`), 1881 days (`General`) and 1052 days (`Quality`) respectively. The considerably smaller timespan for `Seen` documents can be explained through Figure 6: here we plot the distribution of earliest creation timestamps per document. All documents are crawled in 2012 and we observe that the earliest creation timestamps of documents distributed through Twitter are generally quite recent, most of which have been created in 2011 or 2012 — again pointing to the fact that users on Twitter tend to distribute recently created content.

What proportion of content is created over time? Having observed that updates happen across one or more years, we are now concerned with the amount of content created at different points in time. If 99% of a document's content were to be created at the earliest creation timestamp, it would be difficult to argue for adaptations of existing T-IR applications. For this experiment,

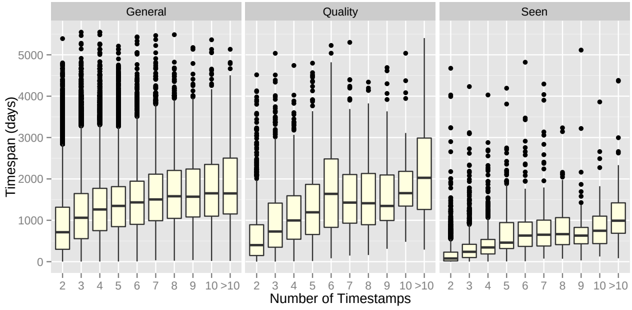


Fig. 5. The document set is partitioned according to the number of creation timestamps (documents with a single creation timestamp are ignored). Shown is the difference (in days) between the oldest and most recent creation timestamp.

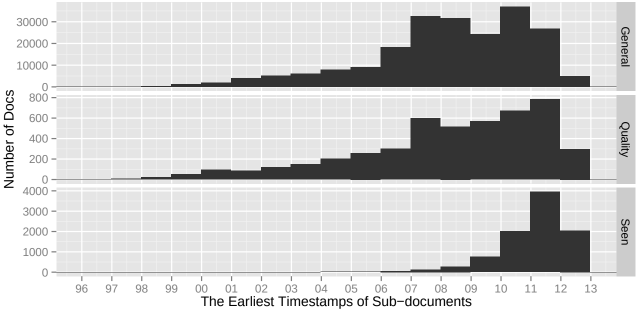


Fig. 6. The earliest sub-document timestamps of different type of Web documents.

we now focus on Web documents with 2, 3 or 4 creation timestamps and compute the percentage of content present in each version. The results are shown in Figure 7. Across the three document sets it holds that the more creation timestamps a document has, the less content is created initially. `Quality` documents have the highest percentage of initially created content across timestamps; one explanation for this observation is the high quality of the content: higher quality leads to more preservation of content over time.

Based on our experiments we conclude that across all 3 document sets, for a significant amount of documents the single-creation-time assumption is wrong.

4.2 Timestamp Inference

Having concluded our exploratory analysis, we now turn to the estimation of sub-document creation timestamps, a mechanism whose accuracy is essential to enable large-scale sub-document timestamping of the Web.

We treat sub-document timestamping as a classification task in line with [23] (ensuring a comparable baseline) and train & test separate models for `Quality`, `Seen` and `General`. Each of our datasets is split into 60% training data, 20%

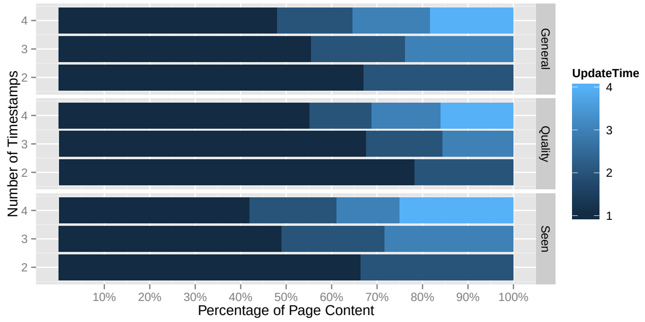


Fig. 7. Overview of content created at different points in time for documents with 2, 3 or 4 creation timestamps. Each bar shows the percentage of content available at each creation timestamp. *UpdateTime* indicates the timestamp if the content created.

validation data and 20% test data. Our **baseline method** is the Random forest (RF) classifier with the same 21 features as in [23]. We first improve the RF with the enlarged feature set (44 features). Subsequently, we employ our 2-stage model (combining RF and CRFs) which employs the RF predictions as features in the CRF.

The algorithms' parameters are tuned by grid search. For the RF classifier, we tune the maximum number of features ($max\_features$) considered for the best split in generating each decision tree. Besides, the number of trees is fixed at 100 without any pruning methods based on some previous work on RF [3, 21]. For CRF, the C-value is tuned from 1×10^{-6} to 1×10^{-4} to adjust the fit of the model.

Lastly, we explore how to exploit the relations between neighbouring sub-documents as part of our CRF models; CRF models that incorporate more neighbours lead to a better prediction, indicating that longer distance dependencies between sub-documents are more helpful than first-order dependency. To avoid an explosion in the number of features though, we only consider neighbourhoods of size four in our experiments.

Timestamping of `Quality` Sub-documents Recall, that we aim to classify the timestamp of individual sub-documents (277,973 sub-documents in `Quality` in total), not individual documents. Due to the skewed nature of `Quality` with less than 1,000 sub-documents created between 1996 and 1998 and more than 69,000 created in 2012, we first balance our dataset in a 5-class setup. For each sub-document we determine the difference in days between the creation time of the sub-document (as found through our IA-based pipeline) and the crawl time of the sub-document as given in the ClueWeb12 metadata. Creating balanced classes yields the following five time intervals: *Class A* represents the interval $(0, 30]$, that is, the sub-documents were created no more than 30 days before they were crawled by the ClueWeb12 crawler. We similarly define the remaining four classes as $B = (30, 365]$, $C = (365, 1095]$, $D = (1095, 2190]$, and $E = (2190, \infty)$. We balance the dataset to test the effectiveness of different temporal inference

Table 1. Sub-document timestamping inference. The baseline is RF with 21 features. Statistically significant changes over RF (all features) are marked ‡ ($p < 0.01$).

		F-Measure / Class				
	Misclassified	A	B	C	D	E
+++ Document set Quality +++						
Baseline method [23]	47.75%	0.55	0.45	0.46	0.46	0.67
RF (44 features)	46.85%	0.55	0.46	0.46	0.47	0.68
2-stage model (RF + CRF) ‡	44.64%	0.59	0.47	0.49	0.50	0.70
+++ Document set Seen +++						
Baseline method	54.37%	0.49	0.44	0.41	0.40	0.54
RF (44 features)	53.49%	0.50	0.44	0.42	0.41	0.55
2-stage model (RF + CRF) ‡	50.30%	0.52	0.49	0.44	0.44	0.60
+++ Document set General +++						
Baseline method	40.36%	0.71	0.55	0.53	0.52	0.63
RF (44 features)	39.36%	0.72	0.56	0.54	0.53	0.64
2-stage model (RF + CRF) ‡	36.70%	0.72	0.59	0.57	0.56	0.69

models, as data imbalance is known to affect some models more than others. Although these developed models cannot be employed as-is in the open Web setting (to timestamp all sub-documents of the Web), they allow us to focus on building sensible models first before tackling the next problem.

The results of these 3 classifier variations[10] on test data are shown in Table 1 (top). The 2-stage approach improves the classification accuracy significantly over our RF baselines[11]. When comparing the two RF classifiers, we observe a slight positive (and statistically significant) change when more features are employed. It means these features are helpful for improving the accuracy in each class, but they can only improve a little.

Timestamping of Seen Sub-documents More than 50% of sub-documents in Seen are timestamped by their ClueWeb12 crawl time, in correspondence with our finding in Section 4.1, that about 38% of Seen documents were crawled by the ClueWeb12 crawler before they were picked up by the IA. Since those sub-documents are not useful for our purposes we ignore them here, and only consider the 306,210 sub-documents in Seen with historical version in IA before the ClueWeb12 crawl time.

We use the same balanced 5-class setup as in the previous experiment with $A = [0, 0]$, $B = (0, 28]$, $C = (28, 152]$, $D = (152, 444]$, and $E = (444, \infty)$. The results of three classifiers[12] in Table 1 (middle) show the same classification trends hold for Seen as for Quality (2-stage model outperforms the RF significantly).

[10] *max_features* is 3 and 6, C-value is 9×10^{-6}

[11] McNemar's test was employed for statistical significance testing, with $p < 0.01$.

[12] *max_features* is 5 and 13, C-value is 9×10^{-5}

However, the accuracies for Seen are all lower than Quality with the same setting. Since the timespans in each class except E of Seen are much smaller than Quality, the accuracy of Seen is acceptable.

Timestamping of General Sub-documents The number of sub-documents in General exceeds six million, which is much larger than Quality and Seen. We first balance our dataset with $A = [0, 0]$, $B = (0, 367]$, $C = (367, 966]$, $D = (966, 1735]$, and $E = (1735, \infty)$. As shown in Table 1 (bottom), the classification accuracy of the three classifiers[13] in General is better than both Quality and Seen. One possible explanation is the larger amount of training data we have available for General. In future work, we will investigate in detail the reasons for this discrepancy.

We conclude that while features have to be selected with care, more relation-aware models (such as CRFs) improve the accuracy of the timestamping process significantly. We find that our trained classifiers do not perform equally well across all balanced classes. This indicates that our current features are more suitable for timestamp inference in a relatively coarse-grained setup, instead of high-resolution time intervals. Based on the observed mis-classification rates (between 37% and 50%) we conclude that we cannot yet employ our pipeline in any application that requires fine-grained and accurate sub-document timestamping.

5 Conclusions

In this work, we have presented a detailed analysis of the content dynamics of Web documents, with the ultimate goal to timestamp their individual sub-documents. We have added significantly to the existing corpus of work, analyzing a data set nearly two magnitudes larger than in previous research. Additionally, we contributed an improved sub-document timestamping inference model and showed its effectiveness across two different Web document sets.

Future work will focus on the improvement of the sub-document timestamping pipeline in order to be able to reliably timestamp all of the Web (or more realistically all of ClueWeb12), which will enable analyses in information diffusion, topic changes, content preservation and other areas.

References

[1] E. Adar, J. Teevan, S. T. Dumais, and J. L. Elsas. The web changes everything: Understanding the dynamics of web content. In *WSDM '09*, pages 282–291, 2009.

[2] R. Baeza-Yates, Á. Pereira, and N. Ziviani. Genealogical trees on the web: a search engine user perspective. In *WWW '08*, pages 367–376. ACM, 2008.

[3] S. Bernard, L. Heutte, and S. Adam. Influence of hyperparameters on random forest accuracy. In *Multiple Classifier Systems*, pages 171–180. Springer, 2009.

[13] $max\_features$ is 7 and 11, C-value is 1×10^{-6}

[4] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*, 47(2):15, 2015.

[5] N. Chambers. Labeling documents with timestamps: Learning from their time expressions. In *ACL '12*, pages 98–106, 2012.

[6] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. 1999.

[7] W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string metrics for matching names and records. In *KDD workshop on data cleaning and object consolidation*, volume 3, pages 73–78, 2003.

[8] G. Cormack, M. Smucker, and C. Clarke. Efficient & effective spam filtering & re-ranking for large web datasets. *Information retrieval*, 14(5):441–465, 2011.

[9] F. de Jong, H. Rode, and D. Hiemstra. Temporal language models for the disclosure of historical text. Royal Netherlands Academy of Arts and Sciences, 2005.

[10] L. Döhling and U. Leser. Extracting and aggregating temporal events from text. In *WWW '14*, pages 839–844, 2014.

[11] T. Ge, B. Chang, S. Li, and Z. Sui. Event-based time label propagation for automatic dating of news articles. In *EMNLP '13*, pages 1–11, 2013.

[12] A. Jatowt, Y. Kawai, H. Ohshima, and K. Tanaka. What can history tell us?: towards different models of interaction with document histories. In *ACM HyperText '08*, pages 5–14, 2008.

[13] A. Jatowt, Y. Kawai, and K. Tanaka. Detecting age of page content. In *Proceedings of the 9th annual ACM international workshop on Web information and data management*, pages 137–144. ACM, 2007.

[14] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems*, 25(3):14, 2007.

[15] N. Kanhabua and K. Nørvåg. Improving temporal language models for determining time of non-timestamped documents. In *Research and advanced technology for digital libraries*, pages 358–370. Springer, 2008.

[16] N. Kanhabua and K. Nørvåg. Using temporal language models for document dating. In *Machine Learning and Knowledge Discovery in Databases*, pages 738–741. 2009.

[17] A. Kumar, M. Lease, and J. Baldridge. Supervised language modeling for temporal resolution of texts. In *CIKM '11*, pages 2069–2072, 2011.

[18] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[19] X. Li and W. B. Croft. Time-based language models. In *CIKM '03*, pages 469–475, 2003.

[20] A. Ntoulas, J. Cho, and C. Olston. What's new on the Web?: the evolution of the Web from a search engine perspective. In *WWW '04*, pages 1–12, 2004.

[21] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. How many trees in a random forest? In *MLDM*, pages 154–168. Springer, 2012.

[22] R. Swan and D. Jensen. Timemines: Constructing timelines with statistical models of word usage. In *KDD Workshop on Text Mining*, pages 73–80, 2000.

[23] Y. Zhao and C. Hauff. Sub-document timestamping of web documents. In *SIGIR '15*, pages 1023–1026, 2015.