Automatic Timeline Summarization from Non-Curated News Streams

Fabián Guevara
University of Amsterdam
ricardo.guevaramelendez@student.uva.nl

Janosch Haber
University of Amsterdam
janosch.haber@student.uva.nl

Joop Pascha
University of Amsterdam
joop.pascha@student.uva.nl

Abstract

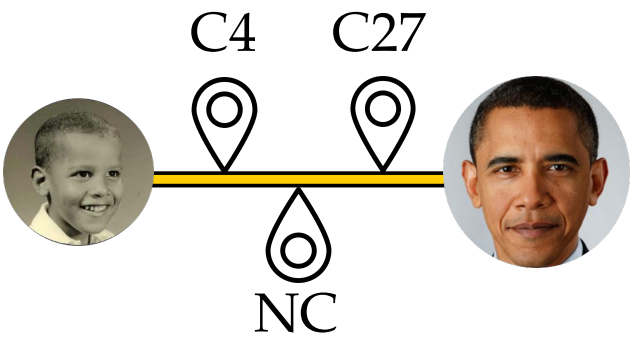
The continuously increasing amount of online news articles requires new ways of filtering relevant information into a human-digestible form. Recently, research has focused on providing such selections by generating timelines for known entities through extending and extracting information from Knowledge Graphs. Contrasting this approach, we propose a new method to generate an entity timeline based directly on a non-curated, unstructured set of news items so as to allow this approach to be extended to long-tail entities.

In this research, Wikipedia pages of entities are seen as a gold-label timeline consisting of information cited from news-worthy articles, while other news articles about those entities that are not cited are treated as negative examples. To learn what makes an article newsworthy, we take a supervised approach based on a set of 28 handcrafted features.

One of our main contributions is a novel, larger dataset for this task, covering 379 unique entities and containing 13146 news articles with an equal distribution of positive and negative examples per entity. Using this dataset we obtain a basic classification accuracy of 68.9% for deciding whether an unseen news article contains relevant information about a given entity. As a baseline method of evaluation, the top article predictions per entity are then summarized and concatenated to generate a dummy Wikipedia entries which we compare to the original ones. As no standardized, gold-label evaluation methods were developed yet, we also propose an A/B testing method for a more qualitative performance estimate.

1. Introduction

Online news coverage heavily increased over the past decade and still continues to grow. This development is caused by two main trends: while the per-day online output of traditional news media outlets is growing, the number of new online news portals is steadily increasing too. As an example, traditional newspaper *New York Times* alone currently publishes 230 online news items per day - an increase of 35% compared to the 170 items in 2010. On the other hand, *Buzzfeed.com*, which was founded in 2006 increased it's monthly output from 900 items in April 2012 to more than 6000 items in April 2016[1].



F1: *A timeline can be seen as a filter of the most important life events of an entity. Here , the news article citations (**C**) in a given entity's Wikipedia page are taken as labeled positive examples of important life events (top), while non-cited news (**NC**) were considered to be negative examples (bottom). A supervised approach is then used to learn what document features discriminate an article's relevance for timeline generation.*

To handle these large quantities of news items and filter them to a human-digestible quantity, two main approaches can currently be found online: Search engines like *Google News*, rank news items based on document features and their source which are sorted per query and time-period that was searched for based on these feature-scores. While this guarantees that most web-crawled documents are processed and filtered, the top results may often not be diverse enough to generate a timeline from. Users interested in a summary of the most interesting set of information facts for a given event or entity may thus end up with a list of different documents about a single event. On the other hand, Knowledge Graphs and Knowledge Bases like online encyclopedia Wikipedia build entity information collections based on the input of a community that manually filters available entity information and produces either a machine or human readable summary. While this guarantees qualitatively descent

[1] https://www.theatlantic.com/technology/archive/2016/05/how-many-stories-do-newspapers-publish-per-day/483845/

and content-diverse entity timelines, these forms of timelines often lack completeness and up-to date information - as well as the possibility of 'zooming in' to a specific period of the entity timeline.

Recently, *Google* augmented their search results with information-cards that are appropriately called *Knowledge Graphs*. We see timelines for entities as a natural progression of this trend of better ways of displaying information. Both of the aforementioned approaches cover a part of the timeline generation problem and provide techniques central to filtering online news media to a appropriately sized human-readable digest. In this paper, we attempt the combine the stronghold of Wikipedia's user-annotated data with Google News more responsive filtering approach of important news coverage.

Our approach is based on three key observations of Fetahu et al. [12], who recently performed an extensive research about the knowledge coverage and temporal lag of entity's Wikipedia articles. Among their results, they discovered that the entity density of Wikipedia slowly converges. This means that reports of newsworthy events over time lead to fewer newly added entity pages because articles about mentioned entities are already present and can simply be extended with the details of the event. In other words, the entity coverage of Wikipedia is converging to completeness. For existing entities, they measured a median lag of about one year, meaning that relevant entity information is mostly added to the entity's Wikipedia page within a year of its publication. And lastly, concerning the source of new information, they show that while the largest part of cited resources are labeled as 'web', on average about 20% of the resources cited in entity pages are newspaper articles from a diverse range of outlets. Based on this, Fetahu et al. [13] present a novel approach to automatically analyze news media content, extracting mentioned entities and determining whether the information it contains should be added to the entities' Wikipedia pages. To do so, they assess a news stream document's internal *salience* and *relative authority* while considering its *novelty* to and *placement* in the entity page.

Interpreting an entity's Wikipedia article as a gold standard, community-curated entity 'biography' timeline, we propose to build on the discovered Wikipedia entity density effect to turn around the analysis process and infer from Wikipedia articles what makes news items suitable for entity timelines. To this end, we collect all news articles used as citations in an entity's Wikipedia article and consider them to be the gold standard set of *positive* examples for *valid* and *relevant* news about a given entity. By featurizing the content of the document with a possible extension of using information from the entity's Wikipedia article itself, we obtain a set of features that are able to determine what makes a news article an appropriate source of timeline-worthy information. We contrast this positive example by adding an equal amount of *negative* examples obtained by querying Google News with the specific entity name in between publication dates of positive samples. Since these articles were not used as references in the Wikipedia article, we assume that they are either irrelevant or less qualitative duplicates of articles in the positive set. With this enriched set of features, supervised methods are applied with the goal of automatically determining (*filtering*) whether an unseen news stream document contains relevant information about tracked entities. Ideally, the resulting filtering method will take salience, novelty and placement into account [13] while obtaining a resulting set of articles that are diversified.

Approach and Contributions. We propose a supervised machine learning approach to automatically determine document relevancy for a given entity. The classification algorithms used are trained on a set of 28 document- and entity-centric features extracted from a gold standard set of *positive* examples obtained from Wikipedia entity pages and an equally large set of *negative* examples sampled from the same news media outlets. The performance of the model's performance is then evaluated in two stages. First, the precision, recall and F1 scores of the binary classification predictions are determined on a held-out test-set of unseen entities to obtain a basic view of model performance. In the second stage, the articles with the top sorted confidence predictions are selected per entity and used for document summarization after which it is compared with the original Wikipedia page by using the ROUGE-metric. If this summary is similar to the entity's Wikipedia article, the assumption is that the obtained timeline is both diverse yet captures articles which are considered highly relevant by the Wikipedia community. In order to get a sense of the relative performance of this score, the same method is applied to randomly selected articles and only the top referenced Wikipedia page articles. The benefits of this approach is that false positive examples can be very similar or even more appriate than positive examples for which this metric accounts for. In summary, we make the following contributions:

- Extend the current timeline generation dataset from Holt et al. [16] with *negative* examples to enable supervised learning.
- Create a novel, richer dataset for the timeline generation problem, containing news article references from the Wikipedia articles of members and election runner-ups of the U.S. *Senate* and *House of Representatives* from 2008 onward.
- Extend a set of long-tail entity features collected by Reinanda et al. [19] to capture the *relevance* of news stream documents.

2. Related Work

The research presented here finds its place at the intersection of a range of open problems. Timeline generation itself is a young problem domain, drawing on previous work in the fields of document filtering, search optimization, entity recognition and automated summarization. To obtain a better overview over what we base our research on, we here present the most relevant and recent developments in those domains.

Knowledge Base Acceleration is a term coined by the 2012 KBA track of the NIST Text REtrieval Conference (TREC). According to the organisation, it .. *seeks to help humans expand knowledge bases like Wikipedia by automatically recommending edits based on incoming content*

streams. This open evaluation measures an automatic system's ability to filter a large stream of text for new knowledge about entities. [2] Kjersten and McNamee [17] provide a baseline for this task by training entity-specific linear Support Vector Machines (SVMs) on sparse binary vectors over more than 100m mentioned words and entities. Subsequently, Balog et al. [5] and Bonnefoy et al. [8] generalize this approach by presenting a set of handcrafted features to determine *central* or *highly relevant* documents from a content stream. Upon analyzing the effects of their features, they conclude that out of their sets of features, word-similarity between a stream document and a respective entity page produces the most predictive features.

Gillick and Dunietz [15] introduce the term *entity salience*, the document intrinsic level of relevance for a given entity. They then demonstrate that through a simple alignment of entity mentions in stream documents with mentions in their accompanying abstracts, a salience label roughly agreeing with manual salience annotations could be created. This allows for a cheap and fast way to automatically create a large set of potentially valid documents for any number of tracked entities. In the same line of work, Reinanda et al. [19] developed and evaluated a method called EIDF for classifying vital and non-vital documents with respect to a given entity. To do so, they designed a set of document-intrinsic features that capture the *informativeness*, *entity salience*, and *timeliness* of news items. All of these features can be assessed in the data stream documents themselves and therefore enables tracking even for long-tail entities without entity pages. Since we attempt to generate an entity timeline based on news stream documents, we will adapt and extend this collection of features to assess document relevance.

Wikipedia Page Generation is the problem of populating Wikipedia pages with content coming from external sources and therefore forms a subdomain of the KBA problem. Initial work in this field was presented by Sauper and Barzilay [21] who learn entity class specific page templates (specifically, the sections of the Wikipedia articles of a certain entity class) from readily populated entity pages and query the internet for documents to fill these templates for novel entities. Fetahu et al. [13] critique that this approach limits generalizability and propose an automated approach for suggesting news articles to update existing Wikipedia entity pages based on their *entity salience*, *relative authority* and *novelty*. Regarding Wikipedia pages as gold standard entity timelines, this research will draw upon the notions of *entity salience*, *relative authority* and *novelty* to determine what information to add to an entity timeline.

Timeline Generation covers the task of automatically generating a timeline out of available entity information. Entity information is either obtained from an existing knowledge graph, or automatically extracted from news stream data. Althoff et al. [4] present an algorithm for the former category, formulating the timeline generation problem in a submodular optimization framework where document relevance, content diversity and temporal diversity of the selected set of documents are optimized jointly. The problem they face, however, is the lack of a gold standard test-set for assessing the performance of their system. To overcome this limitation, they resort to user A/B tests where their production significantly outperforms a global importance baseline.

Using unstructured news stream data, Tran et al. [22] present a novel approach for timeline summarization of high-impact news events. They focus on extracting central entities from relevant documents and dynamically weighting entity salience and document informativeness to improve user experience. Here, too, the product performance was significantly better than baseline performance in both expert and crowd-sourcing assessments. Yang and Mitchell [23] on the other hand cast the problem as a sentence recommendation task, building on a representation learning approach. While this offers a range of interesting approaches for further steps towards an implementation of a timeline application, we will here mostly focus on the conceptual problem of news filtering.

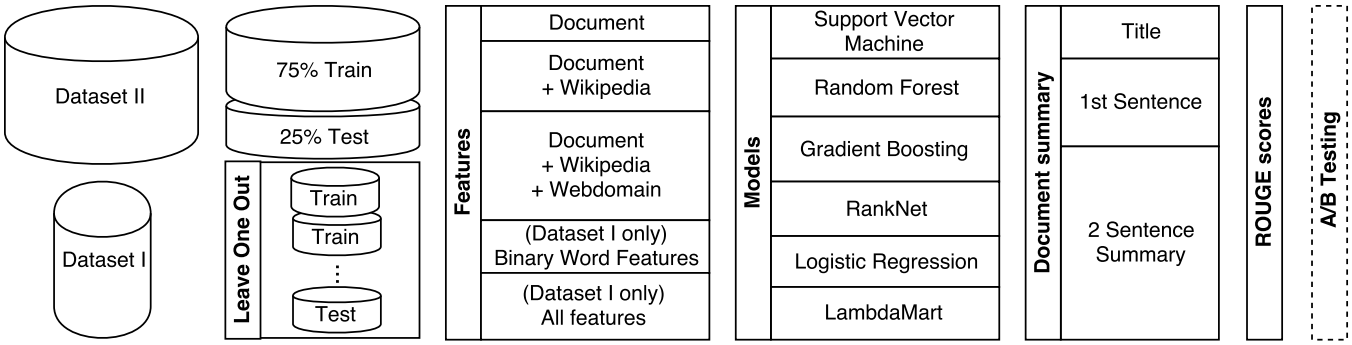
3. Problem Definition

Given a set $\mathcal{E}$ of entities, we define a dataset $\mathcal{D}_e$ consisting of news articles a for each $e \in \mathcal{E}$. $\mathcal{D}_e$ consists of *valid* articles only, that is only articles which cover a single event concerning entity e. The articles are stored with a boolean label l which indicates whether the article is relevant to the entity's timeline. A timeline is defined as a finite list of articles, sorted by publishing date. For any $e \in \mathcal{E}$, the ideal timeline would have every tuple $(article, label) \in \mathcal{D}_e$ such that the label is $TRUE$ and be of a length that is easily processable by a human reader. The objective is to find a model that learns how to build these ideal timelines given a training dataset with annotated articles for several entities.

An essential assumption is that there are underlying characteristics of news articles that make them relevant - independently of the entity it refers to - and that those characteristics can be encoded in a set of features. This means that we are interested in a generalizable model that should not learn entity specific feature weights, but learn what makes an article relevant for any $e \in \mathcal{D}$ instead. Note that in Information Retrieval terminology, an entity can be seen as a query to the database. The critical difference is that every document is known to be about the entity, so the retrieval logic deals exclusively with determining whether a document is relevant or not for a timeline of that entity to guaranty that a timeline for that specific entity can be created.

Since timelines should be short so users can easily obtain the information they contain, determining the best articles to be added to an entity timeline involves a filtering task. point-wise, pair-wise and list-wise methods are used for this. For point-wise methods the document feature representations are used in its entirety (not per query) with a flag indicating if the document originated from Google News (i.e. negative example) or Wikipedia citations (i.e. positive example) functioning as binary prediction target. Filtering the documents is then formulated as choosing the documents with the highest confidence interval to obtain a top-n ranked list that can be seen as a timeline. For pair-wise and list-wise methods, training and testing is applied per entity to

[2] http://trec-kba.org



F2: *Overview of the experimental setup. Starting from the left, for each* ***Dataset*** *a different training approach is taken. Next, different* ***Feature****-sets are created on which 6 different* ***Models*** *are trained. For evaluation 3* ***Document Summary*** *approaches are taken which are then compared to the gold Standard Wikipedia page by obtaining the* ***Rouge scores****. For future work* ***A/B Testing*** *is left.*

form a ranking and then the top documents are sorted on date to form a timeline. These approaches are experimental and as far as we know, no other work has been done on list-wise/pair-wise models for timeline generation. Although list-wise methods are known to outperform the classification techniques in ranking tasks, the task at hand here is not limited to (re-)ranking alone, and therefore point-wise methods should not be discarded by principle.

4. Experimental Setup

Since the timeline generation is a rather new problem domain, the experimental setup of the approaches is strongly based on the current work in Knowledge Base Acceleration and related problems. Here, important characteristics of the document stream items are extracted either through learned or handcrafted features and their weights trained against a gold-label, often human annotated subsection of the document collection. The trained classifiers are then used to predict relevance labels on a held-out test-set to assess the model performance.

What makes timeline generation a more involved task than the document filtering for Knowledge Base Acceleration alone is the fact that (1) timeline quality is a highly subjective measure capturing selection interestingness, timeliness and diversity, which can hardly be expressed numerically and (2) it is concerned with the quality of a *set* of documents which exceeds a simple 'correct' vs. 'incorrect' classification. Among others, this is also one of the main reasons why research struggles to produce a decent gold-standard dataset for the timeline generation problem. Recently, Holt et al. [16] introduced a first dataset especially created for the timeline generation task. It contains links to news article references in the Wikipedia pages of 28 politicians and celebrities, reasoning that those articles could form a gold-standard set of relevant articles for the given entities. They then collect *validity* and *relevance* judgments through crowd-sourcing. The characteristics of this dataset are described in the next section. However, as it contains no negative samples and limited amount of documents (of mostly short-tail entity) to reliably train on, one of our main contributions to the timeline generation problem is the introduction of a method for sampling irrelevant articles to enrich the positive article set and collecting a larger and more solid dataset. Using the same approach as Holt et al. [16], we collected URLs, downloaded and parsed the Wikipedia news references of more than 4000 U.S. politicians which were members or election runner-ups for the U.S. Senate and House of Representatives, yielding a final dataset of 379 entities with a 50-50 positive-negative article collection of more than 4 articles each. An in-depth analysis of this dataset can be found in section 5.

Both of these datasets are used for training and testing in the remainder of this paper. We follow an approach schematically displayed in Figure 2. Using leave-one-out training for dataset I and a 75%-25% test-train split for dataset II, we extract a set of 28 document-centric and entity-centric features from the training documents and train four different classification models proven to be successful in the KBA problem domain. In addition two ranking algorithms were added from the Information Retrieval field to create timeline article selections. For training, we use five different feature sets for dataset I. To save computational resources, only 3 of these were used for dataset II. Model performance is then evaluated on a basic level assessing prediction precision, recall and F1 scores.

In order to rank the output of the classification methods, we use classification confidence to sort positive predictions. The top n aritcles where then taken, where n is the number of cited news articles in the entity's Wikipedia page, to create a timeline summary from. For this, we sort the selection by publishing date and extract either their title, first sentence or two-sentence summary into a new summary document. The same is done for the actual citations in the Wikipedia article. A mockup for how a timeline can look visually can be seen in Figure 20. For both summaries similarity to the Wikipedia article itself is captured through the use of the ROUGE metric [18]. This metric captures how close the selected timeline articles get to the gold-standard 'biography timeline' that is encoded in the Wikipedia article. The following variety of ROUGE metrics were included; ROUGE-N (N-gram comparisons, with unigrams and bigrams), ROUGE-L (longest common subsequence) and ROUGE-SU (Skip-grams plus unigram based).

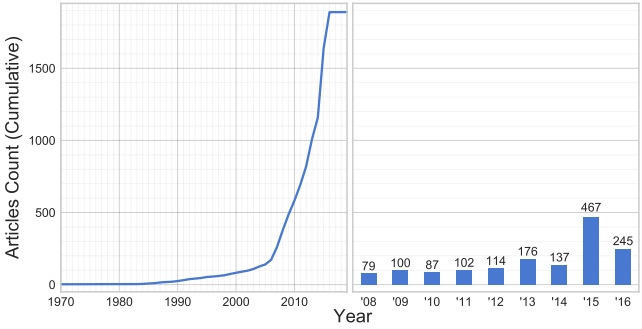
5. Data

Now follows an overview of how the two datasets were assembled and their characteristics to enhance the understanding of algorithm performance in Results.

Dataset I

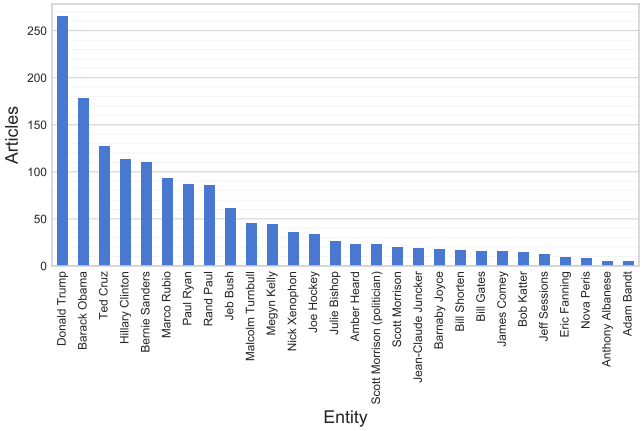
Holt et al. [16] released a dataset called `crowd.csv` containing 3141 document *URL*'s of news articles used as references in the Wikipedia articles of a set of 39 U.S. and Australian politicians together with some celebrities [3]. The dataset contains crowd-sourced annotations for both document *validity* and *relevancy*. Since annotator agreement was only moderate and almost all relevance labels belong to the 'somewhat relevant' class, we decided to not further limit the dataset and use all of the 2570 valid documents.

The python web-scraping Newspaper library [2] was used for webscraping to successfully obtain the content of 1891 of those 2570 articles. This discrepancy was accredited to faulty or no longer active URLs or webdomains for which the parsing library failed. For each article, *publish_date*, *title*, and *content* were extracted. With the library's built-in *nlp*-extension the article's *keywords* and *summary*-sentences were determined by heuristics. Articles that were missing any of the aforementioned fields were removed, leaving a total of 1803 articles. Finally, articles that were published before 2007 were removed due to an elbow-shaped pattern observed in the publication statistics of Wikipedia (see Figure 3), resulting in 1629 articles. Considering that the dataset mainly consist of current politicians (see Figure 4), we expect that this is the main reason for the skewed distribution in Figure 3.



F3: *Total (cumulative) articles per year.* ***Left****: the cumulative article count over all years from a subset of 1891 articles from the URLs provided by Holt et al. [16].* ***Right****: the total amount of articles per year. Dataset I.*

To obtain negative samples for training, we made a closed-world assumption, considering all news articles about the same set of entities that were published during same period of time and not included in the Wikipedia pages to be irrelevant and therefore negative examples. To do so, we determined the date ranges in between the sorted positive examples with a specified cool-down interval to limit the amount of overlapping news articles. A minimum 3-day separation between positive and negative examples seemed reasonable to fulfill this purpose and make sure that the negative samples can indeed be interpreted as negative samples



F4: *Total articles per entity of a subset of 1507 articles taken from URLs provided by Holt et al. [16]. Dataset I.*

with some confidence. We then queried Google News with entity names and date-ranges to obtain URLs to 6356 articles from the top 10 search results. Subsequently, the same webscraping and filtering procedure was applied and the remaining set of 4197 articles was sampled to obtain an exact *50-50* split of negative and positive examples per entity. The negative examples were iteratively sampled by picking the next closest article to the middle of the date ranges as this was hypothesized to further reduce the effect of overlapping news from different news-outlets. In this step also entities which contained fewer negative than positive examples were removed, resulting in a final subset of 1507 positive and an equal amount of negative examples from 28 different entities.

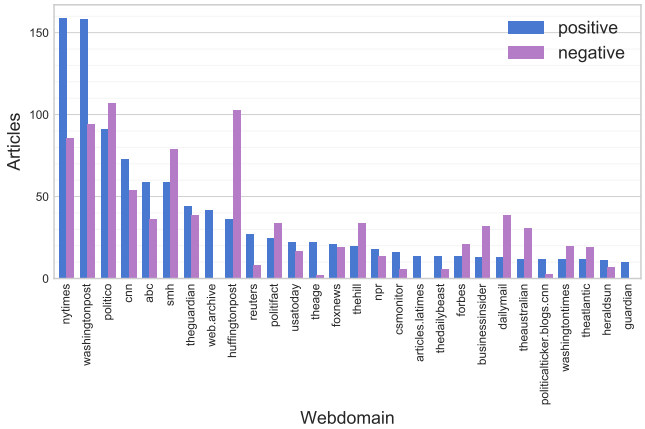
Figure 5 displays the distribution of positive and negative samples per news outlet webdomain, showing considerable differences for most of the sources. We propose that most of these differences can be accredited to a variety of reasons, e.g. the *news outlet respectability*, writing style *formality*, *popularity*, or the *timeliness of publication*. While some of these features can be extracted from the document itself, e.g. a specific writing style that can be seen by the frequency of words, others might not be covered by document-intrinsic features alone. We therefore introduced a boolean webdomain feature to the existing feature set to try to capture this information for classification training (see section 6).

Dataset II

After all filtering and pre-processing steps a collection of documents from 28 different entities remained. Therefore, it was decided to obtain a larger dataset in parallel to further developing the project pipeline. To do so, we followed the same bootstrapping approach as Holt et al. [16], collecting positive article samples from the Wikipedia pages of members and election runner-ups from U.S. Congress, which consists of the House of Representatives and Senate. We limited the dataset to the Congresses from 2008 onward based on the increase in citations (Figure 3) observed in the Holt dataset, which coincides with the increasing amount of articles added on Wikipedia at that same moment [1]. We expect that including articles

T1: *Dataset II statistics. In combination with Table 2 an overview of the main dataset characteristics is given. The values between the two houses in Congress are separated by the '-' sign. Dataset II.*

House of Representatives + Runners Up (6290) - Senators (6968)							
Metric	mean	std	min	25%	50%	75%	max
Sentence doc count	34.6 - 35.6	37.9 - 40.0	1 - 2	16 - 16	26 - 27	41 - 43	666 - 821
Word doc count	891 - 904	934 - 927	90 - 90	416 - 431	686 - 708	1050 - 1093	15.3k - 20.0k
Char per word	4.51 - 4.48	0.267 - 0.257	3.49 - 2.84	4.34 - 4.32	4.52 - 4.49	4.69 - 4.65	6.82 - 6.38
Doc count	26.7 - 49.1	24.9 - 59.5	8 - 8	12 - 18	18 - 28	30 - 50	192 - 318
1st Sentence	31.8 - 31.3	18.4 - 18.4	1 - 2	19 - 18	30 - 29	40 - 40	229 - 200
Title word count	9.77 - 9.76	3.82 - 3.75	1 - 1	7 - 7	9 - 9	12 - 12	41 - 42



F5: *Amount of positive and negative examples per news outlet (≈webdomain). Dataset I.*

from before this period would not represent the current standards of online news and therefore conflict with a number of earlier assumptions about the data. Collecting the dataset was simplified by the fact that Wikipedia has a specific page for all Congress elections labeled */United_States_House_of_Representatives_elections,_XXXX* and */United_States_Senate_elections,_XXXX* where *XXXX* denotes the election year (only even years). From these lists, we gathered a total of 3778 Wikipedia pages for Members or runner-ups for the House of Representatives and 535 for the Senate. From these, a total of 330k reference URLs were obtained, of which 20k were references to articles from news-outlets which the Newspaper library could handle. The positive examples then were obtained similarly to Dataset I (section 5), leaving a set of just 675 politicians after all pre-processing and filtering steps, for which we collected negative samples. Filtering out all entities not fulfilling the 50-50 positive-negative split, the final dataset then consisted of 379 unique entities with a total of 13146 articles (12447 after removing duplicates)

Analysis of dataset II showed no significant differences between the articles from politicians in the House of Representatives and Senate (see Table 2, 1). Therefore it can be concluded that from a general point of view the published news items from both groups are written in a similar writing style. The original ratio of entities of 7 to 1 for House of Representatives and Senate changed significantly, however, the final ratio was close to 1.8 to 1. We expect that this difference is due to the broader coverage of Senators when compared to that of members of the House of Representatives. In addition, the fact that the majority of entities from the House of Commons consisted of runners-up (which were included to obtain a much larger dataset) that never actually made it into Congress was expected to give them significantly lower news coverage and therefore more probable to be removed entirely from the dataset after the validity checks. This is backed up by the fact that the number of authors is more diverse (Table 2).

T2: *Difference between House of Representatives and Senate. The values between the two houses in Congress are separated by the '-' sign. Dataset II.*

House of Representatives + Runners Up (6290) - Senators (6968)					
Metric	entries	entities	authors	publish_dates	urls
Unique	6108 - 6290[3]	229 - 128	3067 - 2864	5235 - 5283	5860 - 6100
Total in dataset	13146	379	5412	9977	12447

This dataset was expected to improve the stability of results by both increased sample size, use-age of only one entity domain, and shift towards a more long-tail set of politicians. A comparison between datasets shows that the total articles per entity was more evenly spread (Figure 7 left, versus Figure 4) and showed less variance over the year of publication (Figure 7 right versus Figure 3 right). Outliers are mostly accredited to politicians who held a higher office after their time in Congress, like most famously former president Barack Obama or former Vice Precident Joe Biden. In addition, the difference in distributions of articles over webdomain between positive and negative examples converged (Figure 5 versus Figure 6), meaning that here the news outlet probably looses its predictive power as classification prior opposed to the one displayed for Dataset I (Figure 5).

Having collected the datasets, the next step in the timeline generation pipeline is encoding the news articles in feature vectors to be used in training. Most of the features require some form of pre-processing described in the next section, followed by a in-detail analysis of our feature sets.

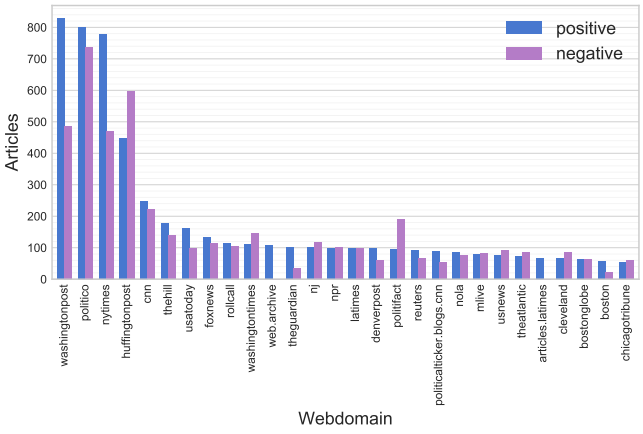
6. Feature Engineering

Pre-processing

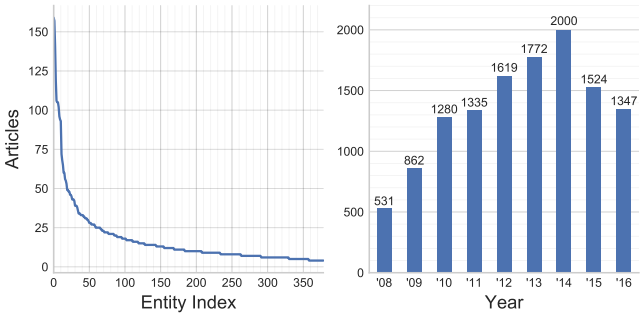
Parsing the articles in the dataset returns document representations as a single string with all HTML markup and special characters preserved. To pre-process this for feature extraction, we first remove newline characters and markup to obtain parsable text. As a next step, entities are extracted by using the *NLTK* python library Tree parser (Bird et al. [7]), which is shown to outperform entity word-tokenizers

T3: *Overview of implemented features.* ***Feature****-column: name of each feature and the parameters it intakes.* ***Description****-column: description of how the features were created.* ***Source****-column: origin of feature.* ***Type****-column: semantic grouping of feature-types.*

#	Feature	Description	Source	Type
1	REL(e)	Number of related entities of e mentioned in the entity's Wikipedia page w_e	[6]	basic
2	DOCREL(a,e)	Fraction of related entity mentions in news article a divided by the total number of entity mentions	[6]	basic
3	NUMFULL(a,e)	Number of full mentions of e in a	[6]	basic
4	NUMPARTIAL(a,e)	Number of partial mentions of e in a	[6]	basic
6	FPOSFULL(a,e)	Index of first full mention of e in a	[6]	basic
6	FPOSPART(a,e)	Index of first partial mention of e in a	[6]	basic
7	LPOSPART(a,e)	Index of last partial mention of e in a	[6]	basic
8	SPREAD(a,e)	Spread (first position - last position) of mentions of e in a	[6]	basic
9	SIM$_{cos}$(a,p_e)	Uni- and bigram cosine similarity between a and w_e	[6]	basic
10	SIM$_{jac}$(a,p_e)	Jaccard similarity between a and w_e	[6]	basic
11	DOCLEN$_{chunk}$(a)	Number of paragraphs in a	[20]	informativeness
12	DOCLEN$_{sent}$(a)	Number of sentences in a	[20]	informativeness
13	PROFILELEN(w_e)	Number of words in w_e	[20]	informativeness
14	NUMMENTIONS(w_e)	Number of entity mentions in w_e	[20]	informativeness
15	FRACENTITIES(a)	Fraction of entity mentions in a divided by article length	[20]	entity salience
16	NUMSENT(a,e)	Number of sentences in a containing entity e	[20]	entity salience
17	FULLFRACT(a,e)	Number of full mentions of e in the article, normalized by number of entity mentions	[20]	entity salience
18	MENTIONFRACT(a,e)	Number of full or partial mentions of e in the article, normalized by number of entity mentions	[20]	entity salience
19	TMATCH$_Y$(a)	Number of year expressions of timestamp t in a	[20]	timeliness
20	TMATCH$_{YM}$(a)	Number of year, month expressions of timestamp t in a	[20]	timeliness
21	TMATCH$_{YMD}$(a)	Number of year, month, date expressions of timestamp t in a	[20]	timeliness
22	QT#_PCT_MENTIONS	Percentage of entity mentions in the #th quantile of a	[8]	other
23	ENTITY_IN_TITLE	Boolean indicating if the entity was mentioned in the title	[8]	other
24	AVG_WORD_LEN(a)	Average word length of a in characters	this paper	formality
25	MENTFRAC(a)	Fraction of mentions of e divided by length of article a in words	this paper	entity salience
26	AVG_SENT_LEN(a)	Average sentence sentence of a in words	this paper	formality
27	WEBDOMAIN	Boolean indicator of the article's news outlet webdomain (considering only top 19 occurring domains)	this paper	other
28	AVG_PAR_LEN(a)	Average paragraph length of a in words	this paper	formality



F6: *Amount of positive and negative examples per news outlet (≈webdomain). Dataset II.*



F7: Left*: total amount of examples per entity.* ***Right*** *the total amount of articles per year. Dataset II.*

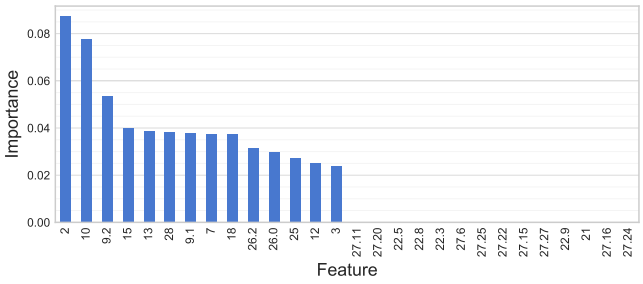
(e.g. Finkel et al. [14]) for little known long-tail entities or datasets with different naming conventions. Some artifacts were introduced in which sub-trees were mistakenly merged together to form one entity, e.g. `Senator John McCain` instead of `John McCain` separately, but was considered within the margin of error. All entity names were then removed from the text and reserved for feature engineering, while the remainder of the article content was then cleaned from stopwords, tokenized, lemmatized and lowercased using NLTK functionality.

Features

As a most basic feature set, these document representations were encoded in binary word vectors per document. This creates sparse vectors of the size of the vocabulary of the collection, which we decided to reduce to a more tractable

size of 250 dimensions using PCA (from 15k and 120k respectively for dataset I and II). This allows to cover a larger part of the collection instead of the small subsection of articles that could be encoded through the most discriminating word space dimensions.

To capture more high-level document information that is not encoded by using solely word-vector representations, we implement a number of document-centric and entity-centric features presented in previous research and extend that collection through a few novel feature types. In table 3, an overview of the implemented features is provided. *Document*-centric features can be extracted from the news article alone, including document statistics as number of words, sentences and paragraphs, writing style difficulty and entity mention counts. This feature class is meant to capture those features that are present in each available news article and therefore work well for long-tail entities. Also, these features should capture the underlaying document characteristics that make a news article relevant - in case they can be determined. The *entity*-centric features on the other hand use additional information from the wntity's Wikipedia page to enrich document details. Features from this class include the set of related entities (other entity mentions in the Wikipedia page of entity e), Wikipedia article statistics and similarity measures between news article and entity Wikipedia page. If in previous literature feature meanings were left ambiguous, we explain our interpretation in more detail. After training, an analysis of the importance of the different features was conducted, see Figure 8 and section 8for more details.



F8: *Feature importance from the best performing Gradient Boosting model. Top 14 and bottom 14 are shown. Notice that some features are dissected into multiple sub-features indicated by a decimal number. Feature numbers correspond with those in Table 3. Dataset I.*

7. Training

For training feature weights, a set of traditional classification methods were used to that were proven to be successful in the Knowledge Base Acceleration domain. Specifically, the basic implementations of Logistic Regression, Random Forests, Gradient Boosting and Support Vector Classification from the `scikit-learn`[4] Machine Learning collection. Model performance was evaluated on a held-out testset that was the same for all datasets and extracted by splitting per entity. The best-performing method was then used to rank articles based on prediction confidence to generate a top n timeline article selection. For the pair-wise/list-wise models, the Lambda Mart and Ranklib implementations from [11] were each trained twice, first optimizing for precision and then in a second run for reciprocal rank for comparison.

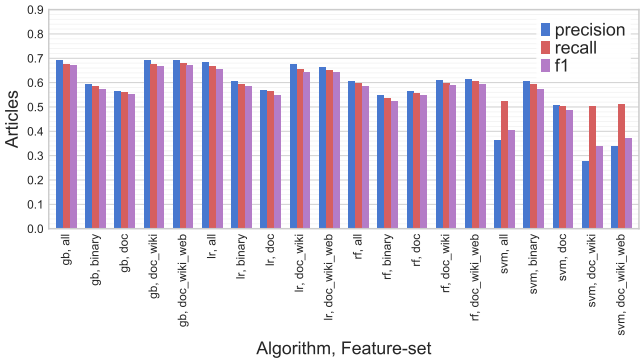
[4] `http://scikit-learn.org/stable/`

8. Results

In the following section, the classification results are split by dataset and compared between the different learning approaches. In a second stage, the top predictions of the best performing timeline summarization model is evaluated.

Dataset I

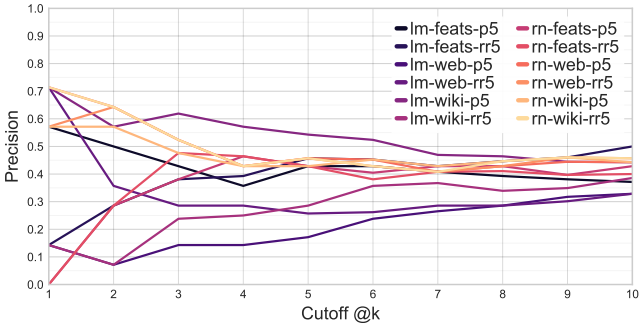
point-wise models For dataset I, all classification models were tested using 5 different feature sets, being (1) top 250 binary word vector document representation, (2) document-centric features only, (3) document-centrtic features plus entity-centric features, (4) document- and entity-centric features plus our experimental webdomain feature and (5) all combined. The results for precision, recall and F1 are summarized in Figure 9.



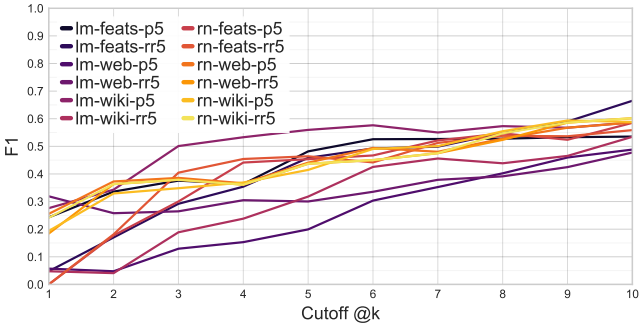
F9: *Comparison of Point-wise classification methods for 3 scoring metrics and 5 feature sets. Algorithms: gb=Gradient Boosting, lr=Logistic Regression, rf=Random Forest, svm=Support Vector Machine. Datasets: doc=document central features, wiki=wikipedia features, web=webdomain features. Dataset I.*

Gradient Boosting outperformed the other models with a precision of nearly 70% and recall and F1 close to that, using either document- and entity-centric features only, document, entity and webdomain features or the entire feature set. Logistic regression is however a close second, exhibiting a similar pattern over the different feature sets. What is remarkable is the SVM model performance: Trained on only the PCA top 250 dimensions of the binary word vector document representation, SVM performs close to the random forest model. Using only the handcrafted features or even adding them to the binary word representations drastically reduces performance. We propose that this effect is due to the weight of the non-normalized features we implemented, forcing the SVM to focus mainly on our handcrafted features which it can not correctly separate in feature-space due to the linear kernel that was used. This is based on the assumption that these features encode non-linear document characteristics. On the other hand, we expect that increasing the dimensionality or changing the compression technique

for the document vector representation will further improve SVM classification performance. Normalization of the features such that their ranges are aligned with the word vector features (0,1) will in our expectation improve results over using just the binary word feature alone.



F10: *Precision @k for Lambda Mart (lm) and RankNet (rn) for the feature selection (feats, web, wiki) with either a precision (p) or reciprocal rank (rr) optimizer @5. Dataset I.*

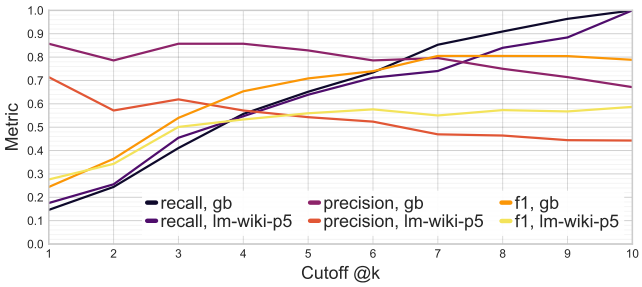


F11: *F1 @k for Lambda Mart (lm) and RankNet (rn) for the feature selection (feats, web, wiki) with either a precision (p) or reciprocal rank (rr) optimizer @5. Dataset I.*

Pair-wise/List-wise models Figure 10 shows the classification precision at cutoff k achieved by Lambda Mart and RankNet. Here, we did not use the binary word vector features for computational reasons for dataset II, leaving us with three different feature sets: (1) document-centric features only, (2) document-centrtic features plus entity-centric features, (3) document- and entity-centric features plus our experimental webdomain feature. Both methods were optimized for either reciprocal rank or precision.

As expected, models optimized for precision outperformed their counterpart optimized for reciprocal rank for a large part of the cutoff graph. With increasing cutoff rank, performance approaches towards the 30%-50% margin for all models. Lambda Mart keeps more variance over time while RankNet performances converge more quickly. Comparing the performance given different feature sets, it can be observed that the entity-centric feature set outperforms both the document-only feature set as well as the full feature set. The webdomain encoding thus apparently negatively influences the predictive power of the feature set for the ranking algorithms. For comparison with the classification models, we will therefore use the LambdaMart instead of the RankNet model with document- and entity-centric features only.

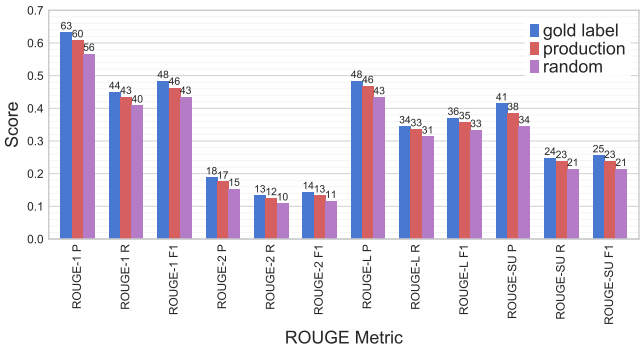
Figure 11 shows the same models evaluated on F1 score. The results indicate that the recall curves develops similarly for all of the tested settings, increasing F1 scores with increasing cutoff rank. For long cutoffs, here the RankNet models appear to start outperforming the LambdaMart scores.



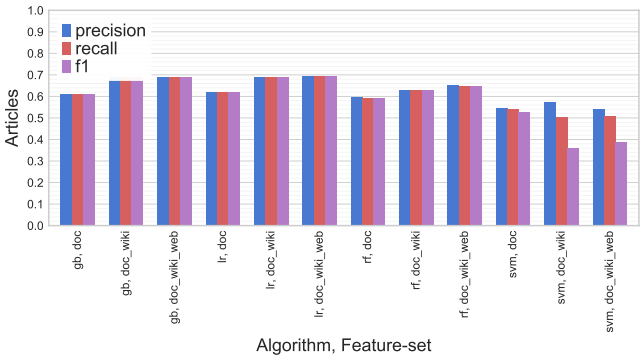
F12: *Precision/recall and f1-score @k comparison for the best point-wise and list/pair-wise method; Gradient Boosting and Lambda Mart. Dataset I.*

Point-wise/List-wise model comparison Now that we determined the best performing settings for both point and pair-wise/list-wise models, we set them against each other in the ranking task. Figure 12 shows the results of this final comparison between Gradient Boosting (all but binary features) and Lambda Mart (document- and entity-centric features only). Gradient boosting here is clearly outperforming on all measures, which is an unexpected result since ranking methods are expected to perform well on this task. We propose two possible explanations for this effect: one reason for the observed difference in performance might be that the feature set is inappropriate for ranking, also indicated by the fact that adding the webdomain features here decreases performance while gradient boosting does not change in performance. If the webdomain feature indeed has a negative effect on the predictability of the correct sample label, gradient boosting appears to be better fit in ignoring these dimensions in the classification task than the ranking methods. Another reason might be an inappropriate encoding of the dataset to train the ranking methods, making it an implementation issue rather than a principled result. Nonetheless, we assume gradient boosting to be our best performing model and use it to create the timeline summary.

Summarization Evaluation To analyze the characteristics of false positives, negative documents labeled as relevant documents by our model, we create a summary of the top n articles of the ranking and compare it to the summary obtained from the n true Wikipedia citations. We measure similarity by the ROUGE score with the actual Wikipedia article to show how close the 'gold-label' summary can indeed get to the gold standard Wikipedia 'biography timeline'. Figure 13 shows the ROUGE scores (1-gram, 2-gram, L and SU*, with precision and recall for each case) for the best Gradient Boosting configuration (production) versus the gold-label summary scores and a random baseline that features n random articles from our 50-50 test-set. It can be observed



F13: *Overview of different ROUGE metric scores by randomly ordering articles (random), our production (Gradient Boosting) and gold label. Dataset I.*



F14: *Comparison of Point-wise classification methods for 3 scoring metrics and 3 feature sets. Algorithms: gb=Gradient Boosting, lr=Logistic Regression, rf=Random Forest, svm=Support Vector Machine. Datasets: doc=document central features, wiki=wikipedia features, web=webdomain features. Dataset II.*

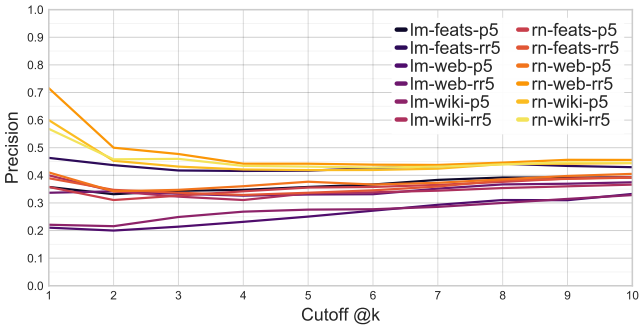
that the obtained production ROUGE scores consistently lie between the gold-label and random summary scores, limited by the fact that the distance between gold-label and random scores are relatively small (between 3% and 5%).

Besides these results, some interesting insights might be gained from this graph: (1) Wikipedia articles either seem to be more than just a short summary of the cited articles, as even gold-label unigram ROUGE scores reach only about 60% - or ROUGE scores are not an appropriate tool to capture timeline summary quality. Note however that not all references are news articles and therefore the Wikipedia article holds more information than the articles in our collection. (2) ROUGE either mostly captures semantic information - or all negative samples are very close to the positive samples. This is based on the fact that random article summary ROUGE scores are only slightly lower than the gold scores. So either ROUGE just measures the semantic similarity of the timeline summary and the actual Wikipedia page and therefore it can be high for negative samples, too - or the events described in the negative samples are so close to the positive ones that summary similarity is almost independent of the articles chosen to be in the timeline. Both observations indicate that ROUGE as an evaluation tool for the task at hand requires a more in-depth analysis.

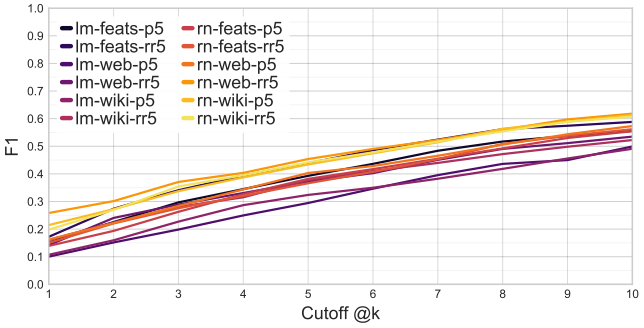
Dataset II

Point-wise models For dataset II, we passed on the binary word vectors as they need some improvement to be of effect and therefore train all models on three different feature sets: (1) document-centric features only, (2) document-centrtic features plus entity-centric features, (3) document- and entity-centric features plus our experimental webdomain feature. Figure 14 shows a comparison of model precision, recall and F1 scores (compare 9). Here logistic regression scores equal to gradient boosting on the full feature set, both reaching 70% precision, recall and F1. While the pattern otherwise is similar to the small dataset, a remarkable difference is that now except for SVM, all settings produce exactly equal precision, recall and F1 scores. We hypothesize that this is an effect of our precise 50-50 split dataset combined with an increased sample-size.

Pair-wise/List-wise models Figures 15 and 16 show the precision and F1 scores, respectively, for Lambda Mart vs. RankNet for the same feature sets as in the point-wise case. Each configuration was trained twice, once optimizing precision at rank 5 and once optimizing reciprocal rank at 5.



F15: *Precision @k for Lambda Mart (lm) and RankNet (rn) for the feature selection (feats, web, wiki) with either a precision (p) or reciprocal rank (rr) optimizer @5. Dataset II.*

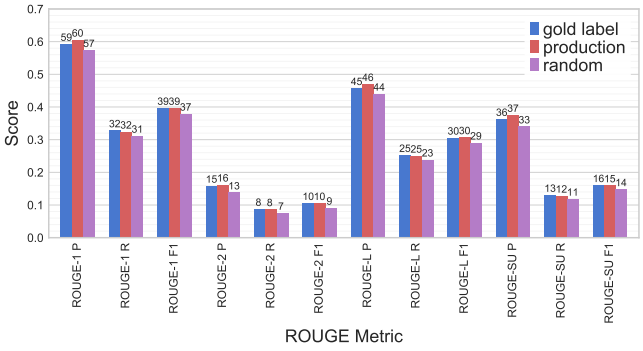


F16: *F1 @k for Lambda Mart (lm) and RankNet (rn) for the feature selection (feats, web, wiki) with either a precision (p) or reciprocal rank (rr) optimizer @5. Dataset II.*

Showing a similar convergence behaviour as their counterparts in dataset I, variance here is greatly decreased, indicating a more stable performance based on the larger sample size. Surprisingly, here RankNet outperforms Lambda Mart (although only minimally at some parts) and while RankNet performs best without the webdomain features, the

by far best performing Lambda Mart model only uses the document-centric features. During testing it was shown that under the same parameter settings the predictions contained mostly the same values indicating that both methods had significant problems with this dataset. A possible explanation for this is that the amount of negative examples were severely limited for this dataset compared to dataset I. During the retrieval of negative examples in between positive examples, ranges in which no negative examples were found were iteratively sampled from bins that contained more than n examples. Therefore, if a lot of date-ranges in between positive examples yield no negative news results, a large chunk of negative examples is formed (due to sorting by date). This only happens for relative short-tail entities that do not have frequent news which shows one of our concerns for obtaining the negative examples for dataset II. For the entities in dataset I which were mostly considered to be long-tail, this was not the case.

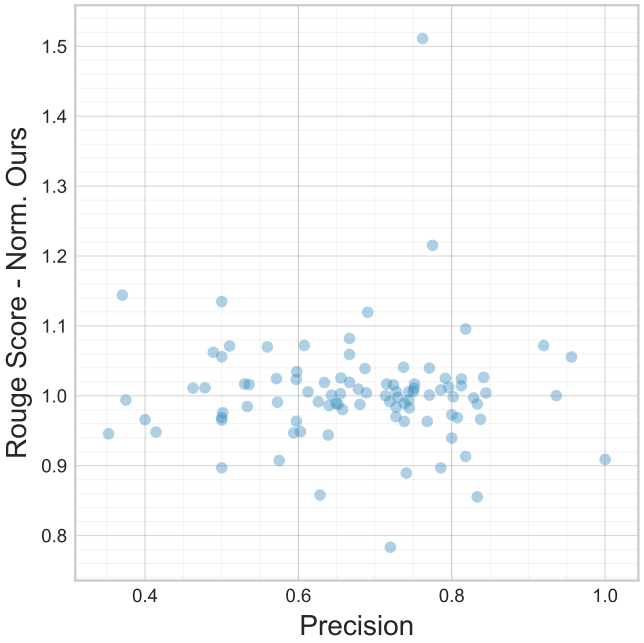
Summarization Evaluation Figure 17 shows the ROUGE scores (1-gram, 2-gram, L and SU*, with precision and recall for each case) for the best Gradient Boosting configuration (production) versus the gold-label summary scores and a random baseline that features n random articles from our 50-50 test-set (compare Figure 13). Remarkably, now our production often even outperforms the gold-label summary. We hypothesize that this stresses the earlier observation that ROUGE might just measure semantic similarity and therefore the model summary can outperform the gold-label summary in similarity with the Wikipedia article, even if precision is lower here.



F17: *Overview of different ROUGE metric scores by randomly ordering articles (random), our production (Gradient Boosting) and gold label. Dataset II.*

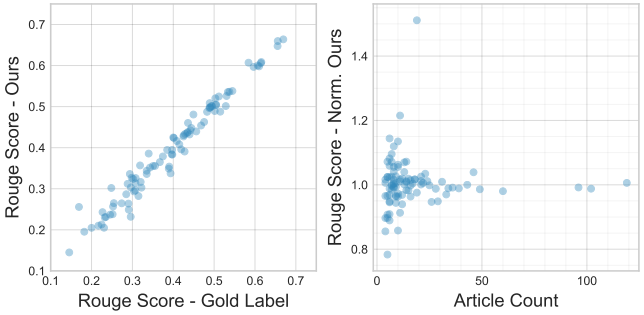
A pair-wise t-test [9] was conducted between the gold-label and production ROUGE scores which showed that the null-hypothesis of the distribution having the same mean could not be rejected (p-value=*0.963*). However, when the production was compared to random or random to gold label the means were found to come from a different distribution (pvalue=4.523e-05 and pvalue=8.342e-05 respectively).

To further analyze the ROUGE performances, Figure 18 shows how our production ROUGE scores normalized by the respective gold-label scores relate to the prediction precision of the classification task. While the sample size is still quite small for generalizable conclusions, it seems that that ROUGE scores remain almost constant against decreasing classification accuracy. This indicates that even an increasingly incorrect set of timeline news articles does not negatively affect the similarity score.



F18: *Relation between precision and normalized ROUGE score (Ours divided by Gold Label per entity). Dataset II.*

Figure 19 shows the correlation between gold-label scores and production scores - indicating a clearly proportional relation between the two measures. An analysis of the normalized model performance for long- vs. short-tail entities therefore also indicates almost no correlation but a constant ROUGE score instead where only variance is higher for long-tail entities (entities with little citations in Wikipedia).



F19: ***Left****: the relation between the Gold Label rouge score and Production Rouge score.* ***Right*** *the relation between article count and normalized ROUGE Scores (Ours divided by Gold Standard per entity). Dataset II.*

9. Discussion

The approach taken here has several limitations. First, the aim was to show the applicability of the current approach to a narrow field of interest, *politics*, which raises questions about the generalizability outside this scope. For instance, Holt et al. [16] included actress Amber Heard for which totally different types of news-events were found newsworthy and therefore translated into poor results in our test-set

(e.g. one of the articles was about smuggling a dog into a country). Secondly, although a significant effort was made to obtain a dataset without biases to specific entities (e.g. anonymization of entity-centric in the document to enhance the model applicability to short-tail entities, 50-50 split per entity) and using evaluation methods that could possibly best denote timeline quality, in the end only human interaction such as A/B testing can best expose this for highly subjective tasks like this.

The current comparison between ROUGE scores which test the similarity between Wikipedia's gold standard timeline with Random and our Production can be improved upon. In our implementation the production and random ROUGE scores as seen in Figure 13 are artificially high as the enforced 50-50 split for both training and testing already imposes that random contains on average 50% of the golden standard articles that are used for summarization. Ideally, in production the evaluation metric measures the semantic similarity between the gold label standard timeline and what was produced. Selecting only negative examples for evaluation yield two important benefits. First, the closed world assumption will be loosened in which all non-positive examples were considered irrelevant. This will allow for more focus on the time-aspect in interests of events as the artificially enforced 3-day separation rule can be dropped. The ROUGE score already allows for this as it is no longer dependent on labels but instead focuses on document similarity. Second, in real world scenario's the vast majority of documents is irrelevant which makes it a more realistic test-set as precision is more important.

Lastly, the fact that search engines put major emphasis on result diversification hints at the possibility that no subjectively *best* timeline exist. Instead it can be expected that there are many possible timelines which all are equally appropriate but are formed around different set characteristics. This makes this task different from for instance content summarization for which the inter-user agreement can be expected to be much higher. Holt et al. [16] demonstrated that the inter-user agreement for news relevance in was considerably low (≈50% for only a 3-label test). The problem of diversification of results is also deeply rooted within our approach as for obtaining the negative results Google News was queried which already diversifies results automatically on many different criteria that share our interest. This is an other reason why the difference between the production and gold label score in Figure 13 can be expected to be relatively low. In order to not rely on Google's page ranking algorithm and obtain a more scientifically sound performance metrics, a controlled set of news outlets can be watched for new news items for a set of entities similar to our approach to obtain an unfiltered and timely access to entity-centric news articles.

10. Conclusion

In this research we showed that our model production timeline's similarity with a gold standard entity biography timeline did not significantly differ from that of a gold-label one, while on the other hand significantly outperforming a random baseline. For similarity comparisons, we used ROUGE-score metrics between timeline summaries extracted from top-ranked news articles and an entity's Wikipedia page. The gold-label summary was obtained from news articles actually cited in the entity's Wikipedia page. While these scores suggest good model performance, no correlation could be shown to classification precision, meaning that ROUGE scores might not be an appropriate quality measure for this task. Nonetheless, one of our main contributions is a much larger and well-analyzed dataset for the timeline generation problem that consists of a significantly larger set of entities than currently available ones.

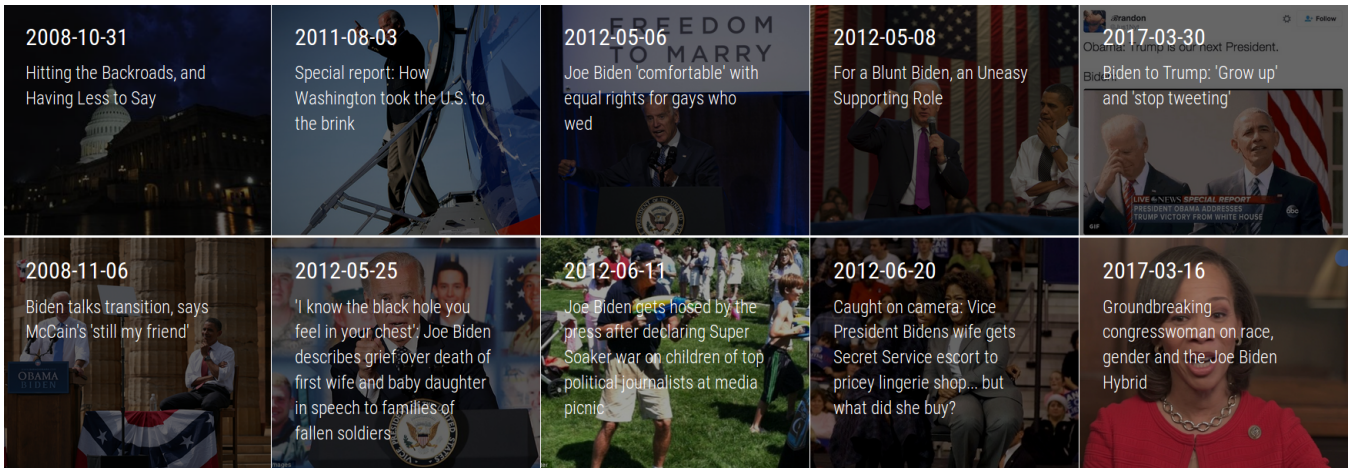
11. Future Work

As ROUGE scores appear to not capture the timeline quality aspects we intended to cover with them, we propose to follow the default evaluation process of the problem domain and obtain models scores through human judgment. To this end we developed an A/B testing application that displays a production timeline and a random timeline in a random order, asking the user to click on the more interesting item (see Figure 20. Click results go directly to a database that can be used to calculate user preferences and significance levels in behaviour differences. Since the scope of this project did not allow to actually conduct this experiment, it is left for future work. We however also want to point out some issues with this approach that made us choose for automatic evaluation in the first place: Titles for example often are not a good representation of an article, diversity and interestingness are hard to evaluate and additional factors like odd-one-out results or distracting images may distort the evaluation process. All of these indicate that even these 'forced' judgments may not be an optimal measure for timeline quality and new evaluation techniques should be investigated.

Concerning model performance, a highly interesting next step would be the implementation of a LSTM to approach the problem of timeline generation, borrowing from Machine Translation trough sentence extraction: Cheng and Lapata [10] used a combination of LSTMs to learn document encodings and extract key sentences based on the hidden states of the encoder module. These extracted sentences are deemed relevant for the document summary as they contain novel information not yet encoded in the memory of the encoder network. Translating this approach to the timeline generation problem, instead of learning the sentence encoding of a single document, the encoder module would learn the document encoding of a set of news articles such that the extractor filters out relevant documents based on the encoder's memory of encountered events. This would cover the timeliness aspect little covered in this report and optimize timelines by guaranteeing the removal of relevant documents that cover similar events. Since dataset I was too small to train a multi-network model like this, we hope that future research can utilize our new dataset to approach this task.

Addressing Concerns on this Solution

There are various concerns to be considered on the practical usage of the solution proposed in this work. They fall on the



F20: *Example of how A/B testing can be used as a feedback mechanism for timeline quality. The top and bottom 5 predictions of the Gradient Boosting algorithm are shown from which the user is asked to click on their preferred timeline. Dataset II.*

following categories:

- **Ethical**: The personal impact of the solution is relegated to the data used. As this is a filtering task, no new information is added, so any personal damage can only result due to the data in its pure form, and never because of any output of this solution.
- **Legal and Accountability**: the solution is meant to work only with publicly available data. In fact, since it is meant for news outlets articles, the information is as public as it can be, by principle. Regarding responsibility for any faults, the fact this deals with filtering makes it pretty safe (as wouldn't be the case if new data were generated). This puts a boundary of worst case scenarios below that of the input data being published.
- **Software Quality**: Most of the libraries used are very standard for their particular purpose and are in fact included with most common python installations (e.g. *pandas* for data handling, *numpy* and *sklearn* for scientific, mathematical and statistical manipulation, *request* for http requests, *beautifulsoup* for HTML scraping) and of highly widespread use. One of the least common libraries we heavily relied on, *newspaper* (a news scraper), even if faulty, wouldn't lead to consequences beyond keeping us from getting an even higher dataset.
- **Security**: This approach is meant to work with newsworthy entities with publicly available data, so no data leaking could result in any harm that wouldn't occur with the original data alone

References

[1] History of wikipedia - wikipedia. `https://en.wikipedia.org/wiki/History_of_Wikipedia`. (Accessed on 07/01/2017).

[2] codelucas/newspaper: News, full-text, and article metadata extraction in python 3. `https://github.com/codelucas/newspaper`. (Accessed on 06/17/2017).

[3] xavi-ai/tlg-dataset: Dataset of news articles for the timeline generation problem. `https://github.com/xavi-ai/tlg-dataset`. (Accessed on 06/17/2017).

[4] Tim Althoff, Xin Luna Dong, Kevin Murphy, Safa Alai, Van Dang, and Wei Zhang. Timemachine: Timeline generation for knowledge-base entities. *CoRR*, abs/1502.04662, 2015. URL `http://arxiv.org/abs/1502.04662`.

[5] Krisztian Balog, Heri Ramampiaro, Naimdjon Takhirov, and Kjetil Nørvåg. Multi-step classification approaches to cumulative citation recommendation. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, OAIR '13, pages 121–128, Paris, France, France, 2013. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE. ISBN 978-2-905450-09-8. URL `http://dl.acm.org/citation.cfm?id=2491748.2491775`.

[6] Krisztian Balog, Heri Ramampiaro, Naimdjon Takhirov, and Kjetil Nørvåg. Multi-step classification approaches to cumulative citation recommendation. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, pages 121–128. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2013.

[7] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[8] Ludovic Bonnefoy, Vincent Bouvier, and Patrice Bellot. A weakly-supervised detection of entity central documents in a stream. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 769–772, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484180. URL `http://doi.acm.org/10.1145/2484028.2484180`.

[9] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[10] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. *CoRR*, abs/1603.07252, 2016. URL http://arxiv.org/abs/1603.07252.

[11] W Bruce Croft, Jamie Callan, J Allan, C Zhai, D Fisher, TT Avrahami, T Strohman, D Metzler, P Ogilvie, M Hoy, et al. The lemur project. *Center for Intelligent Information Retrieval, Computer Science Department, University of Massachusetts Amherst*, 140, 2013.

[12] Besnik Fetahu, Abhijit Anand, and Avishek Anand. How much is wikipedia lagging behind news? *CoRR*, abs/1703.10345, 2017. URL http://arxiv.org/abs/1703.10345.

[13] Besnik Fetahu, Katja Markert, and Avishek Anand. Automated news suggestions for populating wikipedia entity pages. *CoRR*, abs/1703.10344, 2017. URL http://arxiv.org/abs/1703.10344.

[14] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[15] Dan Gillick and Jesse Dunietz. A new entity salience task with millions of training examples. In *Proceedings of the European Association for Computational Linguistics*, 2014.

[16] Xavier Holt, Will Radford, and Ben Hachey. Presenting a new dataset for the timeline generation problem. *arXiv preprint arXiv:1611.02025*, 2016.

[17] Brian Kjersten and Paul McNamee. The hltcoe approach to the trec 2012 kba track. Technical report, TREC '12. DTIC Document, 2012.

[18] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073465. URL http://dx.doi.org/10.3115/1073445.1073465.

[19] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. Document filtering for long-tail entities. *CoRR*, abs/1609.04281, 2016. URL http://arxiv.org/abs/1609.04281.

[20] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. Document filtering for long-tail entities. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 771–780. ACM, 2016.

[21] Christina Sauper and Regina Barzilay. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 208–216, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9. URL http://dl.acm.org/citation.cfm?id=1687878.1687909.

[22] Nam Khanh Tran, Tuan Tran, and Claudia Niederée. *Beyond Time: Dynamic Context-Aware Entity Recommendation*, pages 353–368. Springer International Publishing, Cham, 2017. ISBN 978-3-319-58068-5. doi: 10.1007/978-3-319-58068-5_22. URL http://dx.doi.org/10.1007/978-3-319-58068-5_22.

[23] Bishan Yang and Tom Mitchell. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, 2016.