A Temporally Sensitive Submodularity Framework for Timeline Summarization

Sebastian Martschat[*]
Knowledge Architecture & Innovation
BASF SE
67056 Ludwigshafen am Rhein, Germany
sebastian.martschat@basf.com

Katja Markert
Institute of Computational Linguistics
Heidelberg University
69120 Heidelberg, Germany
markert@cl.uni-heidelberg.de

Abstract

Timeline summarization (TLS) creates an overview of long-running events via dated daily summaries for the most important dates. TLS differs from standard multi-document summarization (MDS) in the importance of date selection, interdependencies between summaries of different dates and by having very short summaries compared to the number of corpus documents. However, we show that MDS optimization models using submodular functions can be adapted to yield well-performing TLS models by designing objective functions and constraints that model the temporal dimension inherent in TLS. Importantly, these adaptations retain the elegance and advantages of the original MDS models (clear separation of features and inference, performance guarantees and scalability, little need for supervision) that current TLS-specific models lack. An open-source implementation of the framework and all models described in this paper is available online.[1]

1 Introduction

There is an abundance of reports on events, crises and disasters. *Timelines* (see Table 1) summarize and date these reports in an ordered overview. *Automatic Timeline Summarization* (TLS) constructs such timelines from corpora that contain articles about the corresponding event.

In contrast to standard *multi-document summarization* (MDS), in TLS we need to explicitly model the temporal dimension of the task, specifically we need to select the most important dates for a long-running event and summarize each of these dates. In addition, TLS deals with a much larger number of documents to summarize,

2011-03-16
Security forces break up a gathering in Marjeh Square in Damascus of 150 protesters holding pictures of imprisoned relatives. Witnesses say 30 people are arrested.
2011-03-24
President Bashar al-Assad orders the formation of a committee to study how to raise living standards and lift the law covering emergency rule, in place for 48 years.
2011-03-29
Government resigns.

Table 1: Excerpt from a Syrian War Reuters timeline.

enhancing scalability and redundancy problems. These differences have significant consequences for constraints, objectives, compression rates and scalability (see Section 2.2).

Due to these differences, most work on TLS has been separate from the MDS community.[2] Instead, approaches to TLS start from scratch, optimizing task-specific heuristic criteria (Chieu and Lee, 2004; Yan et al., 2011b; Wang et al., 2016, inter alia), often with manually determined parameters (Chieu and Lee, 2004; Yan et al., 2011b) or needing supervision (Wang et al., 2016). As features and architectures are rarely reused or indeed separated from each other, it is difficult to assess reported improvements. Moreover, none of these approaches give performance guarantees for the task, which are possible in MDS models based on function optimization (McDonald, 2007; Lin and Bilmes, 2011) that yield state-of-the art models for MDS (Hong et al., 2014; Hirao et al., 2017).

In this paper we take a step back from the differences between MDS and TLS and consider the following question: *Can MDS optimization models be expanded to yield scalable, well-performing TLS models that take into account the temporal properties of TLS, while keeping MDS advantages*

[*]Work conducted while the author was a researcher at the Institute of Computational Linguistics, Heidelberg University.

[1]http://smartschat.de/software

[2]The TLS systems in (Yan et al., 2011b; Tran et al., 2013a) are compared to some simple MDS systems as baselines, but not to state-of-the art ones.

Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 230–240
Brussels, Belgium, October 31 - November 1, 2018.

such as modularity and performance guarantees? In particular, we make the following contributions:

- We adapt the submodular function model of Lin and Bilmes (2011) to TLS (Section 3). This framework is scalable and modular, allowing a "plug-and-play" approach for different submodular functions. It needs little supervision or parameter tuning. We show that even this straightforward MDS adaptation equals or outperforms two strong TLS baselines on two corpora for most metrics.
- We modify the MDS-based objective function by adding temporal criteria that take date selection and interdependencies between daily summaries into account (Section 4).
- We then add more complex temporal constraints, going beyond the simple cardinality constraints in MDS (Section 5). These new constraints specify the uniformity of the timeline daily summaries and date distribution. We also give the first performance guarantees for TLS using these constraints.
- We propose a TLS evaluation framework, in which we study the effect of temporal objective functions and constraints. We show performance improvements of our temporalizations (Section 6). We also present the first oracle upper bounds for the problem and study the impact that timeline properties, such as compression rates, have on performance.

2 Timeline Summarization

Given a query (such as *Syrian war*) TLS needs to (i) extract the most important events for the query and their corresponding dates and (ii) obtain concise daily summaries for each selected date (Allan et al., 2001; Chieu and Lee, 2004; Yan et al., 2011b; Tran et al., 2015a; Wang et al., 2016).

2.1 Task Definition and Notation

A *timeline* is a sequence $(d_1, v_1), \ldots, (d_k, v_k)$ where the d_i are dates and the v_i are summaries for the dates d_i. Given are a query q and an associated corpus C that contains documents relevant to the query. The task of *timeline summarization* is to generate a timeline t based on C. The number of dates in t as well as the length of the daily summaries are typically controlled by the user.

We denote with U the set of sentences in C. We assume that each sentence in U is dated (either by a date expression appearing in the sentence or by the publication date of the article it appears in). For a sentence s we write $d(s)$ for the date of s.

2.2 Relation to MDS

In MDS, we also need to generate a (length-limited) summary of texts in a corpus C (with an optional query q used to retrieve the corpus). In the traditional DUC multi-document summarization tasks[3], most tasks are either not event-based at all or concentrate on one single event. In contrast, in TLS, the corpus describes an event that consists of several subevents that happen on different days.

This difference has substantial effects. In MDS, criteria (such as coverage and diversity) and length constraints apply on a global level. In TLS, the whole summary is naturally divided into per-day summaries. Criteria and constraints apply on a global level as well as on a per-day level.

Even for the small number of DUC tasks that do focus on longer-running events, several differences to TLS still hold. First, the temporal dimension in the DUC gold standard summaries and system outputs is playing a minor role, with few explicit datings of events and a non-temporal structure of the output, leading again to the above-mentioned differences in constraints and criteria. The ROUGE evaluation measures used in MDS (Lin, 2004) also do not take into account temporality and do not explicitly penalize wrong datings. Second, corpora in TLS typically contain thousands of documents per query (Tran et al., 2013b, 2015a). This is magnitudes larger than the corpora usually considered for MDS (Over and Yen, 2004). This leads to a low compression rate[4] and requires approaches to be scalable.

3 Casting TLS as MDS

In the introduction, we identified several issues in existing TLS research, including lack of modularity, insufficient separation between features and model, and the lack of performance guarantees. Global constrained optimization frameworks used in MDS (McDonald, 2007; Lin and Bilmes, 2011) do separate constraints, features and inference and allow for optimal solutions or solutions with performance guarantees. They also can be used in an unsupervised manner. We now cast TLS as MDS, employing constraints and criteria used for stan-

[3] https://duc.nist.gov/

[4] *Compression rate* is the length of the summary divided by the length of the source (Radev et al., 2004).

dard MDS (Lin and Bilmes, 2011). While this ignores the temporal dimension of TLS, it will give us a baseline and a starting point for systematically incorporating temporal information.

3.1 Problem Statement and Inference

We can understand summarization as an optimization of an objective function that evaluates sets of sentences over constraints. Hence, let U be a set of sentences in a corpus and let $f\colon 2^U \to \mathbb{R}_{\geq 0}$ be a function that measures the quality of a summary. Let $\mathcal{I} \subseteq \{X \mid X \in 2^U\}$ be a set of constraints[5]. We then consider the optimization problem

$$S^* = \arg\max_{S \subseteq U, S \in \mathcal{I}} f(S). \quad (1)$$

Solving Equation 1 exactly does not scale well (McDonald, 2007) and is therefore inappropriate for the large-scale data used in TLS. The greedy Algorithm 1 that iteratively constructs an output solves the equation approximately (also used in McDonald (2007) and Lin and Bilmes (2011)).

Algorithm 1 Greedy algorithm.

Input: A set of sentences U, a function f, a set of constraints $\mathcal{I}$

function GREEDY(U, f, $\mathcal{I}$)

Set $S = \emptyset$, $K = U$

while $K \neq \emptyset$ **do**

$s = \arg\max_{t \in K} f(S \cup \{t\}) - f(S)$

if $S \cup \{s\} \in \mathcal{I}$ **then**

$S = S \cup \{s\}$

$K = K \setminus \{s\}$

Output: A summary S

3.2 Monotonicity and Submodularity

The results obtained by GREEDY can be arbitrarily bad. However, there are performance guarantees if the objective function f and the constraints $\mathcal{I}$ are "sufficiently nice" (Calinescu et al., 2011). Many results rely on objective functions that are *monotone* and *submodular*. A function f is monotone if $A \subseteq B$ implies that $f(A) \leq f(B)$. A function f is submodular if it possesses a "diminishing returns property", i.e. if for $A \subseteq B \subset U$ and $v \in U \setminus B$ we have $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$.

From now on we assume that the function f is of the form $f \equiv \sum_{i=1}^{m} f_i$ with monotone submodular $f_i\colon U \to [0, 1]$ ($i \in \{1, \ldots, m\}$). We normalize all f_i to $[0, 1]$. By closure properties of monotonicity and submodularity, f is also submodular.

3.3 MDS Constraints

Constraints help to define a summary's structure, and the performance guarantee of the greedy algorithm depends on them. In MDS, typical constraints are upper bounds in the number of sentences or words, corresponding to cardinality ($|S| \leq m$) or knapsack constraints ($\sum_{s \in S} |\text{words}(s)| \leq m$) for some upper bound m. When optimizing a submodular monotone function under such constraints, GREEDY has a performance guarantee of ≈ 0.63 and ≈ 0.39 respectively (Calinescu et al., 2011; Lin and Bilmes, 2011). That is, for cardinality constraints, the output is at least 0.63 as good as the optimal solution in terms of objective function value.

3.4 MDS Objective Functions

In MDS, approaches typically try to maximize coverage and diversity. In its simplest form, Lin and Bilmes (2011) model coverage as

$$f_{\text{Cov}}(S) = \sum_{s \in S} \sum_{v \in U} \text{sim}(s, v), \quad (2)$$

where $\text{sim}\colon U \times U \to \mathbb{R}_{\geq 0}$ is a sentence similarity function, e.g. cosine of word vectors.

Lin and Bilmes (2011) model diversity via

$$f_{\text{Div}}(S) = \sum_{i=1}^{k} \sqrt{\sum_{s \in P_i \cap S} r(s)} \quad (3)$$

where $P_1, \ldots, P_k$ is a partition of U (e.g. obtained by semantic clustering) and $r\colon U \to \mathbb{R}_{\geq 0}$ is a singleton reward function. We get diminished reward for adding additional sentences from one cluster.

3.5 Application to TLS

Applying this MDS model to TLS as-is may not be adequate. For example, since the length constraints only limit the total number of sentences, some days in the timeline could be overrepresented. Furthermore, if objective functions ignore temporal information, we may not be able to extract sentences that describe very important events lasting only for short time periods. Instead, natural units for TLS are both the whole timeline as well as individual dates, so criteria and constraints for TLS should accommodate both units.

[5] An example are length constraints, which can be expressed as $\mathcal{I} = \{X \mid |X| \leq m, X \in 2^U\}$ for some m.

4 Temporalizing Objective Functions

We now systematically add temporal information to the objective function by (i) temporalizing coverage functions, (ii) temporalizing diversity functions, and (iii) adding date selection functions. We prove the monotonicity and submodularity of all functions in the supplementary material.

4.1 Temporalizing Coverage

MDS coverage functions (Equation 2) ignore temporal information, computing coverage on a corpus-wide level. We temporalize them by modifying the similarity computation. This is a minimal but fundamental modification. Previous work in TLS noted that coverage for candidate summaries for a day d should look mainly at the temporally local neighborhood, i.e. at sentences whose dates are close to d (Chieu and Lee, 2004; Yan et al., 2011b). We investigate two variants of this idea. The first uses a hard cutoff (Chieu and Lee, 2004), restricting similarity computations to sentences that are at most p days apart:

$$\text{sim}_p(s,t) = \begin{cases} \text{sim}(s,t) & |d(s) - d(t)| \leq p \\ 0 & |d(s) - d(t)| > p \end{cases} \quad (4)$$

The second uses a soft variant (Yan et al., 2011b). Let $g\colon \mathbb{N} \to \mathbb{R}_{>0}$ be monotone with $g(0) = 1$. We set $\text{sim}^g(s,t) = \text{sim}(s,t)/g(|d(s) - d(t)|)$. Thus, all date differences are penalized, and greater date differences are penalized more.

4.2 Temporalizing Diversity

As with coverage, standard MDS diversity functions (Equation 3) ignore temporal information. If the singleton reward r in f_{Div} relies on sim, as is the case with many implementations, then temporalizing sim implicitly temporalizes diversity. We now go beyond such an implicit temporalization.

In TLS, we want to apply diversity on a temporal basis: we do not want to concentrate the summary on very few, albeit important dates, but we want date (and subevent) diversity. f_{Div}, however, typically uses only a semantic criterion to obtain a partition, e.g. by k-means clustering of sentence vector representations (Lin and Bilmes, 2011). This may wrongly conflate events, such as two unrelated protests on different dates. We can instead employ a temporal partition. The simplest method is to partition the sentences by their date, i.e. for a temporalized diversity function f_{TempDiv} we have the same form as in Equation 3, but P_i contains all sentences with date d_i, where $d_1, \ldots, d_k$ are all sentence dates.

4.3 Date Selection Criteria

An important part of TLS is *date selection*. Dedicated algorithms for date selection use frequency and patterns in date referencing to determine date importance (Tran et al., 2015b). Most date importance measures can be integrated into the objective function to allow for joint date selection and summary generation.[6] One well-performing date selection baseline is to measure for each date how many sentences refer to it. This objective can be described by the monotone submodular function

$$f_{\text{DateRef}}(S) = \sum_{d \in d(S)} \left|\{u \in U \mid u \text{ refers to } d\}\right|.$$

4.4 Combining Criteria

We combine coverage, diversity and date importance via unweighted sums for our final objective functions. An alternative would be to combine them via weighted sums learned from training data (Lin and Bilmes, 2011, 2012) but since there are only few datasets available for training and testing TLS algorithms we choose the unweighted sum to estimate as few parameters as possible from data.

5 Temporalizing Constraints

The MDS knapsack/cardinality constraints are too simple for TLS as an overall sentence limit does not constrain a timeline to have daily summaries of roughly similar length or enforce other uniformity properties. We introduce constraints going beyond simple cardinality, and prove performance guarantees of GREEDY under such constraints.

5.1 Definition of Constraints

Typically, we have two requirements on the timeline: the total number of days should not exceed a given number ℓ and the length of the daily summary (in sentences) should not exceed a given number k (for every day). Let d be the function that assigns each sentence its date. For a set $S \subseteq U$, the requirements can be formalized as

$$\left|\{d(s) \mid s \in S\}\right| \leq \ell \quad (5)$$

and, for all $s \in S$,

$$\left|\{s' \mid s' \in S,\ d(s') = d(s)\}\right| \leq k. \quad (6)$$

[6] Our framework can also be extended to accommodate pipelined date selection. We leave this to future work.

5.2 Performance Guarantees

While the constraints expressed by Equations 5 and 6 are more complex than constraints used in MDS, they have a property in common: if a set S fulfills the constraints (i.e. $S \in \mathcal{I}$), then also any subset $T \subseteq S$ fulfills the constraints (i.e. $T \in \mathcal{I}$). In combinatorics, such constraints are called *independence systems* (Calinescu et al., 2011).

Definition 1. *Let V be some set and $\mathcal{I} \subset 2^V$ be a collection of subsets of V. The tuple $(V, \mathcal{I})$ is called an* independence system *if (i) $\emptyset \in \mathcal{I}$ and (ii) $B \in \mathcal{I}$ and $A \subseteq B$ implies $A \in \mathcal{I}$.*

Optimization theory shows that GREEDY also has performance guarantees when generalizing cardinality/knapsack constraints to "sufficiently nice" independence systems. Based on these results, we prove Lemma 1 (see the suppl. material):

Lemma 1. *Let $\mathcal{I}$ be the set of subsets of U that fulfill Equations 5 and 6. Then* GREEDY *has a performance guarantee of $1/(k+1)$.*

The lemma implies that for small k that is typical in TLS (e.g. $k = 2$), we obtain a good approximation with reasonable constraints. However, our performance guarantees are still weaker than for MDS (for example, 0.33 for $k = 2$ compared to 0.63 in MDS). The reason for this is that our constraints are more complex, going beyond the simple well-studied cardinality and knapsack constraints. We also observe that this is a worst-case bound: in practice the performance of the algorithm may approach the exact solution (as Lin and Bilmes (2010) show for MDS). However, such an analysis is out of scope for our paper, since computing the exact solution is intractable in TLS.[7]

6 Experiments

We evaluate the performance of modeling TLS as MDS and the effect of various temporalizations.

6.1 Data and Preprocessing

We run experiments on *timeline17* (Tran et al., 2013b) and *crisis* (Tran et al., 2015a). Both data sets consist of (i) journalist-generated timelines on events such as the Syrian War as well as (ii) corresponding corpora of news articles on the topic scraped via Google News. They are publically available[8] and have been used in previous work (Wang et al., 2016).[9] Table 2 shows an overview.

Name	Topics	TLs	Docs	Sentences	
				Total	Filtered
timeline17	9	19	4,622	273,432	56,449
crisis	4	22	18,246	689,165	121,803

Table 2: Data set statistics.

No	Start	End	Dates	Avg. Daily Summ. Length
1	2010-04-20	2010-05-02	13	4
2	2010-04-20	2012-11-15	16	2
3	2010-04-20	2010-10-15	12	2
4	2010-04-20	2010-09-19	48	2
5	2010-04-20	2011-01-06	102	3

Table 3: Properties for the *BP oil spill* timelines in *timeline17*. The corpus contains documents for 218 dates from 2010-04-01 to 2011-01-31.

In the data sets, even timelines for the same topic have considerable variation. Table 3 shows properties for the five *BP oil spill* timelines in *timeline17*. There is substantial variation in range, granularity and average daily summary length.

Following previous work (Chieu and Lee, 2004; Yan et al., 2011b), we filter sentences in the corpus using keywords. For each topic we manually define a set of keywords. If any of the keywords appears in a sentence, the sentence is retained.

We identify temporal expressions with HeidelTime (Strötgen and Gertz, 2013). If a sentence s contains a time expression that can be mapped to a day d via HeidelTime we set the date of s to d (if there are multiple expressions we take the first one). Otherwise, we set the date of s to the publication date of the article which contains s.[10]

6.2 Evaluation Metrics

Automatic evaluation of TLS is done by ROUGE (Lin, 2004). We report ROUGE-1 and ROUGE-2 F_1 scores for the *concat*, *agreement* and *align+ m:1* metrics for TLS we presented in Martschat and Markert (2017). These metrics perform evaluation by concatenating all daily summaries, evaluating only matching days and evaluating aligned

[7] McDonald (2007) and Lin and Bilmes (2010) already report scalability issues for obtaining exact solutions for MDS, which is of smaller scale and has simpler constraints than our task.

[8] http://www.l3s.de/~gtran/timeline/

[9] The datasets used in Chieu and Lee (2004) or Nguyen et al. (2014) are not available.

[10] This procedure is in line with previous TLS work (Chieu and Lee, 2004). The focus of the current paper is not on further improving date assignment.

dates based on date and content similarity, respectively. We evaluate date selection using F_1 score.

6.3 Experimental Settings

TLS has no established settings. Ideally, reference and predicted timelines should be given the same compression parameters, such as overall length or number of days.[11] Since there is considerable variation in timeline parameters (Table 3), we evaluate against each reference timeline individually, providing systems with the parameters they need via extraction from the reference timeline, including range and needed length constraints. We set m to the number of sentences in the reference timeline, ℓ to the number of dates in the timeline, and k to the average length of the daily summaries.

Most previous work uses different or unreported settings, which makes comparison difficult. For instance, Tran et al. (2013b) do not report how they obtain timeline length. Wang et al. (2015, 2016) create a constant-length summary for each day that has an article in the corpus, thereby comparing reference timelines with few days with predicted timelines that have summaries for each day.

6.4 Baselines

Past work on *crisis* generated summaries from headlines (Wang et al., 2016) or only used manual evaluation (Tran et al., 2015a). Past work on *timeline17* evaluates with ROUGE (Tran et al., 2013b; Wang et al., 2016) but suffers from the fact that parameters for presented systems, baselines and reference timelines differ or are not reported (see above). Therefore, we reimplement two baselines that were competitive in previous work (Yan et al., 2011b; Wang et al., 2015, 2016).

Chieu. Our first baseline is CHIEU, the unsupervised approach of Chieu and Lee (2004). It operates in two stages. First, it ranks sentences based on similarity: for each sentence s, similarities to all sentences in a 10-day window around the date of s are summed up[12]. This yields a ranked list of sentences, sorted by highest to lowest summed up similarities. Using this list, a timeline containing one-sentence daily summaries is constructed as follows: iterating through the ranked sentence list, a sentence is added to the timeline depending on the *extent* of the sentences already in the timeline. Extent of a sentence s is defined as the smallest window of days such that the total similarity of s to sentences in this window reaches at least 80% of the similarity to the sentences in the full 10-day window. If the candidate sentence does not fall into the extent of any sentence already in the timeline, it is added to the timeline.

As we can see, the model and parameters such as daily summary length are intertwined in this approach. We therefore reimplement CHIEU exactly instead of giving it reference timeline parameters. As we describe below, we use the same sentence similarity function as Chieu and Lee (2004).

Regression. Our second baseline is REG, a supervised linear regression model (Tran et al., 2013b; Wang et al., 2015). We represent each sentence with features describing its length, number of named entities, unigram features, and averaged/summed tf-idf scores. During training, for each sentence, standard ROUGE-1 F_1 w.r.t. the reference summary of the sentence's date is computed. The model is trained to predict this score.[13] During prediction, sentences are selected greedily according to predicted F_1 score, respecting temporal constraints defined by the reference timeline.

6.5 Model Parameters

For all submodular models and for CHIEU we use sparse inverse-date-frequency sentence representations (Chieu and Lee, 2004)[14]. This yields a vector representation v_s for each sentence s. We set $\text{sim}(s,t) = \cos(v_s, v_t)$. We did not tune any further parameters but re-used settings from previous work. For modifications to sim when temporalizing coverage and diversity (Section 4), we use a cutoff of 10 (as Chieu and Lee (2004)), and consider $g(x) = \sqrt{x+1}$ for reweighting. We choose the square root since it quickly provides strong penalizations for date differences but then saturates. Following Lin and Bilmes (2011), we set singleton reward for f_{Div} to $r(s) = \sum_{u \in U} \text{sim}(s,u)$ and obtain the partition $P_1, \ldots, P_k$ by k-means clustering with $k = 0.2 \cdot |U|$. We obtain a temporalization f_{TempDiv} of diversity by considering a partition of sentences induced by their dates (see Section 4).

[11] This would mirror settings in MDS, where reference and predicted summary have the same length constraint.

[12] This corresponds to the *Interest* ranking proposed by Chieu and Lee (2004). We do not use the more complex *Burstiness* measure since *Interest* was found to perform at least as well in previous work when evaluated with ROUGE-based measures (Wang et al., 2015, p.c.)

[13] We use per-topic cross-validation (Tran et al., 2013b).

[14] In preliminary experiments, results using such sparse representations were higher than results using dense vectors.

6.6 Results

Results are displayed in Table 4. The numbers are averaged over all timelines in the respective corpus. We test for significant differences using an approximate randomization test (Noreen, 1989) with a p-value of 0.05.

Baselines. Overall, performance on *crisis* is much lower than on *timeline17*. This is because (i) the corpora in *crisis* contain articles for more days over a larger time span and (ii) average percentage of article publication dates for which a summary in a corresponding reference timeline exists is 11% for *timeline17* and 3% for *crisis*. This makes date selection more difficult. On *crisis*, CHIEU outperforms REG except for date selection. On *timeline17*, REG outperforms CHIEU for four out of seven metrics. Timelines in *crisis* contain fewer dates and shorter daily summaries than timelines in *timeline17*, which aligns well with CHIEU's redundancy post-processing.

TLS as MDS. The model ASMDS uses standard length constraints from MDS and an objective function combining non-temporalized f_{Cov} and f_{Div}. It allows us to evaluate how well standard MDS ports to TLS. Except for *concat* and *date selection* on *crisis*, this model outperforms both baselines, while providing the advantages of modularity, non-supervision and feature/inference separation discussed throughout the paper.

Temporalizing Constraints. The model TLSCONSTRAINTS uses the temporal constraints described in Section 5, but has the same objective function as ASMDS. Compared to ASMDS, there are improvements on all metrics on *timeline17* and similar performance on *crisis*.

Temporalizing Criteria. We temporalize ASMDS objective functions (Section 4) via modifications of the similarity function (cutoffs/reweightings), replacing diversity by temporal diversity f_{TempDiv}, and adding date selection f_{DateRef}. Constraints are kept non-temporal. If modifications improve over ASMDS we also check for cumulative improvements. Modifying similarity is not effective, results drop or stay roughly the same according to most metrics. The other modifications improve performance w.r.t. most metrics, especially for date selection.

Temporalizing Constraints and Criteria. Lastly, we evaluate the joint contribution of temporalized constraints and criteria.[15] Modifications to the similarity function have a positive effect, especially reweighting. f_{DateRef} provides information about date importance not encoded in the constraints, improving results on *crisis*.

Oracle Results. Previous research in MDS computed oracle upper bounds (e.g. Hirao et al. (2017)). To estimate TLS difficulty and our limitations, we provide the first oracle upper bound for TLS: For each sentence s, we compute ROUGE-1 F_1 g_s w.r.t. the reference summary for the sentence's date. We then run GREEDY for $f_{\text{Oracle}}(S) = \sum_{s \in S} g_s$, employing the same constraints as TLSCONSTRAINTS (see Table 7).

Scores of the models are most similar to oracle results for the temporally insensitive *concat* metric, with gaps comparable to gaps in MDS (Hirao et al., 2017). The biggest gap is in *date selection* F_1. This also leads to higher differences in the scores of temporally sensitive metrics, highlighting the importance of temporal information.

6.7 Analysis

We now investigate where and how temporal information helps compared to ASMDS. We have already identified two potential weaknesses of modeling TLS as MDS: the low compression rate (Section 2) and the likely case that ASMDS overrepresents certain dates in a timeline (Section 3). We now analyze the behavior of AsMDS w.r.t. these points and discuss the effect of temporal information. To avoid clutter, we restrict analysis to *timeline17* and report only *align+ m:1* ROUGE-1 F_1.

Effect of Compression Rate. We hypothesize that difficulty increases as compression rate decreases. We measure compression rate in two ways. We first adopt the definition from MDS and define *corpus compression rate* as the number of sentences in a reference timeline divided by the number of sentences in the (unfiltered) corresponding corpus. Second, we define a TLS-specific notion called *spread* as the number of dates in the reference timeline divided by the maximum possible number of dates given its start and end date. For example, the timeline from Table 1 in the introduction has spread $3/14$. We see that timelines with lowest compression rate/spread are indeed the hardest (Table 5). Temporal information leads to improvements in all categories.

[15] We do not evaluate f_{TempDiv}, since the temporal constraints already capture temporal diversity.

Model	concat R1	concat R2	agree R1	agree R2	align+ m:1 R1	align+ m:1 R2	Date Sel. F_1
timeline17							
Baselines							
CHIEU	0.296	0.072	0.039	0.016	0.066	0.019	0.251
REG	0.336	0.065	0.063	0.014	0.074	0.016	0.491
Non-temporal Submodular Models							
ASMDS	0.351†	0.088*	0.071†	0.019	0.086†	0.022	0.452†
Temporalizing Constraints							
TLSCONSTRAINTS	0.368†	0.090$^{\dagger *}$	0.082$^{\dagger *}$	0.022	0.098$^{\dagger *}$	0.025*	0.482†
Temporalizing Criteria							
ASMDS+cutoff	0.338^{x}	0.083*	0.065†	0.021	0.077	0.024	0.393$^{\dagger *x}$
ASMDS+reweighting	0.329^{x}	0.081^{x}	0.063†	0.019	0.075^{x}	0.022	0.390$^{\dagger *x}$
ASMDS+f_{DateRef}	0.357†	**0.092**$^{\dagger *x}$	0.082$^{\dagger *x}$	0.022*	0.095$^{\dagger *x}$	0.025*	0.529$^{\dagger x}$
ASMDS+f_{TempDiv}	0.347	0.088*	0.088$^{\dagger *x}$	0.026$^{\dagger *}$	0.103$^{\dagger *x}$	0.029$^{\dagger *x}$	0.526$^{\dagger x}$
ASMDS+f_{TempDiv}+f_{DateRef}	0.347	0.090*	**0.092**$^{\dagger *x}$	**0.027**$^{\dagger *x}$	0.105$^{\dagger *x}$	**0.030**$^{\dagger *x}$	**0.544**$^{\dagger *x}$
Temporalizing Constraints and Criteria							
TLSCONSTRAINTS+cutoff	0.366†	0.085*	0.091$^{\dagger *x}$	0.023*	0.105$^{\dagger *x}$	0.026*	0.505$^{\dagger x}$
TLSCONSTRAINTS+reweighting	**0.371**†	0.088$^{\dagger *}$	0.091$^{\dagger *x}$	0.026$^{\dagger *x}$	**0.106**$^{\dagger *x}$	0.028$^{\dagger *x}$	0.506$^{\dagger x}$
TLSCONSTRAINTS+f_{DateRef}	**0.371**$^{\dagger *x}$	0.090$^{\dagger *}$	0.089$^{\dagger *x}$	0.023*	0.103$^{\dagger *x}$	0.026*	0.517$^{\dagger x}$
TLSCONSTRAINTS+f_{DateRef}+reweighting	0.370$^{\dagger *}$	0.091$^{\dagger *}$	0.090$^{\dagger *x}$	0.024*	0.104$^{\dagger *x}$	0.027*	0.515$^{\dagger x}$
crisis							
Baselines							
CHIEU	**0.374**	0.070	0.029	0.008	0.052	0.012	0.142
REG	0.271	0.034	0.014	0.001	0.028	0.003	0.189
Non-temporal Submodular Models							
ASMDS	0.309$^{\dagger *}$	0.064*	0.037*	0.009*	0.060*	0.014*	0.183†
Temporalizing Constraints							
TLSCONSTRAINTS	0.339$^{\dagger *x}$	0.066*	0.035*	0.008*	0.058*	0.012*	0.180†
Temporalizing Criteria							
ASMDS+cutoff	0.283$^{\dagger x}$	0.061$^{\dagger *}$	0.036*	0.011*	0.050*	0.014*	0.186
ASMDS+reweighting	0.294$^{\dagger *}$	0.061$^{\dagger *}$	0.039*	0.011*	0.056*	0.015*	0.212$^{\dagger *}$
ASMDS+f_{DateRef}	0.314$^{\dagger *}$	0.067*	0.042$^{\dagger *}$	0.009*	0.065$^{\dagger *x}$	0.014*	0.248$^{\dagger *x}$
ASMDS+f_{TempDiv}	0.311†	0.062*	0.034*	0.007*	0.058*	0.012^{*x}	0.196$^{\dagger *}$
ASMDS+f_{TempDiv}+f_{DateRef}	0.311$^{\dagger *}$	0.064*	0.039$^{\dagger *}$	0.008*	0.063$^{\dagger *}$	0.012*	0.233$^{\dagger *x}$
Temporalizing Constraints and Criteria							
TLSCONSTRAINTS+cutoff	0.323$^{\dagger *x}$	0.068*	0.046$^{\dagger *}$	0.011*	0.066$^{\dagger *}$	0.015*	0.242$^{\dagger x}$
TLSCONSTRAINTS+reweighting	0.332$^{\dagger *x}$	0.071^{*x}	0.044$^{\dagger *}$	0.009*	0.068$^{\dagger *}$	0.014*	0.237$^{\dagger x}$
TLSCONSTRAINTS+f_{DateRef}	0.333$^{\dagger *x}$	0.069^{*x}	0.045$^{\dagger *x}$	0.009*	0.067$^{\dagger *x}$	0.013*	0.248$^{\dagger *x}$
TLSCONSTRAINTS+f_{DateRef}+reweighting	0.333$^{\dagger *x}$	**0.072**^{*x}	**0.054**$^{\dagger *x}$	**0.012**$^{\dagger *}$	**0.075**$^{\dagger *x}$	**0.016***	**0.281**$^{\dagger *x}$

Table 4: Results. Highest values per column/dataset are boldfaced. For the submodular models, † denotes sign. difference to CHIEU, * to REG, ^{x} to ASMDS.

(Over)representation of Dates. We hypothesized that ASMDS may overrepresent certain dates. We test this hypothesis by measuring the length (in sentences) of the longest daily summary in a timeline, and computing mean and median over all timelines (Table 6). The numbers confirm the hypothesis: When modeling TLS as MDS, some daily summaries tend to be very long. By construction of the constraints employed, the effect does not occur or is much weaker for CHIEU, REG and TLSCONSTRAINTS. Temporal objective functions (as in ASMDS+f_{TempDiv}+f_{DateRef}) also weaken the effect substantially.

7 Related Work

The earliest work on TLS is Allan et al. (2001), who introduce the concepts of usefulness (conceptually similar to coverage) and novelty (similar to diversity), using a simple multiplicative combination. However, both concepts are not temporalized. The notion of usefulness is developed further as "interest" by Chieu and Lee (2004), which we use as one of our baselines. Chieu and Lee (2004) compute interest/coverage in a static local date-based window, instead of using global optimization as we do. They handle redundancy only during post-processing s.t. the interplay between coverage and diversity is not adequately modeled. Further optimization criteria are intro-

Name	Compression rate r			Spread s		
	$r \in [0, 0.001]$	$r \in (0.001, 0.01]$	$r \in (0.01, 0.1]$	$s \in [0, 1/3]$	$s \in (1/3, 2/3]$	$s \in (2/3, 1]$
CHIEU	0.06	0.08	0.07	0.06	0.08	0.04
REG	0.04	0.09	0.07	0.05	0.11	0.11
ASMDS	0.05	0.10	0.09	0.07	0.10	0.10
TLSCONSTRAINTS	0.08	0.10	0.10	0.08	0.12	0.14
ASMDS+f_{TempDiv}+f_{DateRef}	0.09	0.11	0.12	0.09	0.13	0.13

Table 5: Results (*align+ m:1* ROUGE-1 F_1) by *compression rate* and *spread* on timeline17.

Name	Max. Length	
	Mean	Median
Reference	5.6 ± 2.7	5
CHIEU	1.0 ± 0.0	1
REGRESSION	2.3 ± 1.7	2
ASMDS	23.7 ± 41.2	8
TLSCONSTRAINTS	2.3 ± 1.7	2
ASMDS+f_{TempDiv}+f_{DateRef}	3.8 ± 5.3	1

Table 6: Length of longest daily summary, mean and median over all timelines on *timeline18*.

	concat		agree		align+ m:1		Date
Corpus	R1	R2	R1	R2	R1	R2	F_1
tl17	0.50	0.18	0.30	0.14	0.30	0.14	0.87
crisis	0.49	0.16	0.34	0.14	0.35	0.14	0.95

Table 7: Oracle results optimizing per-day R1 F_1.

duced by Yan et al. (2011b,a) and Nguyen et al. (2014), but their frameworks suffer from a lack of modularity or from an unclear separation of features and architecture. Wang et al. (2015) devise a local submodular model for predicting daily summaries in TLS, but they do not model the whole timeline generation as submodular function optimization under suitable constraints.

Wang et al. (2016) tackle only the task of generating daily summaries without date selection using a supervised framework, greedily optimizing per-day predicted ROUGE scores, using images and text. In contrast, Kessler et al. (2012) and Tran et al. (2015b) only tackle date selection but do not generate any summaries. We consider the full task, including date selection and summary generation.

TLS is related to standard MDS. We discussed differences in Section 2. Our framework is inspired by Lin and Bilmes (2011) who cast MDS as optimization of submodular functions under cardinality and knapsack constraints. We go beyond their work by modeling temporally-sensitive objective functions as well as more complex constraints encountered in TLS.

A related task is TREC *real-time summarization* (RTS) (Lin et al., 2016).[16]. In contrast to TLS, this task requires *online* summarization by presenting the input as a stream of documents and emphasizes novelty detection and lack of latency. In addition, RTS focuses on social media and has a very fine-grained temporal granularity. TLS also has an emphasis on date selection and dating for algorithms and evaluation which is not present in RTS as the social media messages are dated a priori.

8 Conclusions

We show that submodular optimization models for MDS can yield well-performing models for TLS, despite the differences between the tasks. Therefore we can port advantages such as modularity and separation between features and inference, which current TLS models lack. In addition, we temporalize these MDS-based models to take into account TLS-specific properties, such as timeline uniformity constraints, importance of date selection and temporally sensitive objectives. These temporalizations increase performance without losing the mentioned advantages. We prove that the ensuing functions are still submodular and that the more complex constraints still retain performance guarantees for a greedy algorithm, ensuring scalability.

Acknowledgments

We thank the anonymous reviewers and our colleague Josef Ruppenhofer for feedback on earlier drafts of this paper.

References

James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of new topics. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in*

[16] Predecessors of this task were the *update* and *temporal summarization tasks* (Aslam et al., 2015)

Information Retrieval, New Orleans, Louis., 9–12 September 2001, pages 49–56.

Javed A. Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Testuya Sakai. 2015. TREC 2015 temporal summarization track overview. In *Proceedings of the Twenty-Fourth Text REtrieval Conference,* Gaithersburg, Md., 17–20 November 2015.

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766.

Hai Leong Chieu and Yoong Keok Lee. 2004. Query based event extraction along a timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* New York, N.Y., 25–29 July 2004, pages 425–432.

Tsutomu Hirao, Masaaki Nishino, Jun Suzuki, and Masaaki Nagata. 2017. Enumeration of extractive oracle summaries. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, volume 1: Long Papers,* Valencia, Spain, 3–7 April 2017, pages 386–396.

Kai Hong, John M. Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the 9th International Conference on Language Resources and Evaluation,* Reykjavik, Iceland, 26–31 May 2014, pages 1608–1616.

Remy Kessler, Xavier Tannier, Carloine Hagège, Véronique Moriceau, and André Bittar. 2012. Finding salient dates for building thematic timelines. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Jeju Island, Korea, 8–14 July 2012, pages 730–739.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Text Summarization Branches Out Workshop at ACL '04,* Barcelona, Spain, 25–26 July 2004, pages 74–81.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Proceedings of Human Language Technologies 2010: The Conference of the North American Chapter of the Association for Computational Linguistics,* Los Angeles, Cal., 2–4 June 2010, pages 912–920.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Portland, Oreg., 19–24 June 2011, pages 510–520.

Hui Lin and Jeff Bilmes. 2012. Learning mixtures of submodular shells with application to document summarization. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence,* Catalina Island, CA, USA, 14–18 July 2012, pages 479–490.

Jimmy Lin, Adam Roegiest, Luchen Tan, Richard McCreadie, Ellen Voorhees, and Fernando Diaz. 2016. Overview of the TREC 2016 real-time summarization track. In *Proceedings of the Twenty-Fifth Text REtrieval Conference, 2016*.

Sebastian Martschat and Katja Markert. 2017. Improving ROUGE for timeline summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers,* Valencia, Spain, 3–7 April 2017, pages 285–290.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of the European Conference on Information Retrieval,* Rome, Italy, 2-5 April 2007, pages 557–564.

Kiem-Hieu Nguyen, Xavier Tannier, and Véronique Moriceau. 2014. Ranking multidocument event descriptions for building thematic timelines. In *Proceedings of the 25th International Conference on Computational Linguistics,* Dublin, Ireland, 23–29 August 2014, pages 1208–1217.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses. An Introduction*. Wiley, New York.

Paul Over and James Yen. 2004. An introduction to DUC 2004: Intrinsic evaluation of generic news text summarization systems. In *Proceedings of the 2004 Document Understanding Conference held at the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Boston, Mass., 6–7 May 2004, pages 1–21.

Dragomir R. Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40:919–938.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015a. Timeline summarization from relevant headlines. In *Proceedings of the 37th European Conference on Information Retrieval,* Vienna, Austria, 29 March – 2 April 2015, pages 245–256.

Giang Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013a. Predicting relevant news events for timeline summaries. In *Proceedings of the 22nd World Wide Web Conference,* Rio de Janeiro, Brasil, 13–17 May, 2013, pages 91–92.

Giang Tran, Eelco Herder, and Katja Markert. 2015b. Joint graphical models for date selection in timeline summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Beijing, China, 26–31 July 2015, pages 1598–1607.

Giang Tran, Tuan Tran, Nam-Khanh Tran, Mohammad Alrifai, and Nattiya Kanhabua. 2013b. Leveraging learning to rank in an optimization framework for timeline summarization. In *Proceedings of the SIGIR 2013 Workshop on Time-aware Information Access (TAIA-13),* Dubling, Ireland, 1 August 2013.

Lu Wang, Claire Cardie, and Galen Marchetti. 2015. Socially-informed timeline generation for complex events. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Denver, Col., 31 May – 5 June 2015, pages 1055–1065.

William Yang Wang, Yashar Mehdad, Dragomir R. Radev, and Amanda Stent. 2016. A low-rank approximation approach to learning joint embeddings of news stories and images for timeline summarization. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* San Diego, Cal., 12 – 17 June 2016, pages 58–68.

Rui Yan, Liang Kong, Congrui Huang, Xiajun Wan, Xiaoming Li, and Yan Zhang. 2011a. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing,* Edinburgh, Scotland, U.K., 27–29 July 2011, pages 433–443.

Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaming Li, and Yan Zhang. 2011b. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Beijing, China, 25–29 July 2011, pages 745–754.