Unsupervised Timeline Generation for Wikipedia History Articles

Sandro Bauer and **Simone Teufel**
Computer Laboratory
University of Cambridge
Cambridge, United Kingdom
{sandro.bauer, simone.teufel}@cl.cam.ac.uk

Abstract

This paper presents a generic approach to content selection for creating timelines from individual history articles for which no external information about the same topic is available. This scenario is in contrast to existing works on timeline generation, which require the presence of a large corpus of news articles. To identify salient events in a given history article, we exploit lexical cues about the article's subject area, as well as time expressions that are syntactically attached to an event word. We also test different methods of ensuring timeline coverage of the entire historical time span described. Our best-performing method outperforms a new unsupervised baseline and an improved version of an existing supervised approach. We see our work as a step towards more semantically motivated approaches to single-document summarisation.

1 Introduction

While there has been much work on generating history timelines automatically, these approaches are commonly evaluated on events that took place in recent decades, as they depend on the availability of large numbers of articles describing the same historical period. If such a rich data source is available, it is possible to exploit document creation times, redundancy across documents, as well as back-references to earlier events in order to identify salient events. For instance, the start of the Iraq War in 2003 is mentioned frequently in a general news corpus, including in articles published years after the event took place. The high number of mentions suggests that the beginning of the Iraq War was an important historical event.

However, for most historical periods covered in history articles (e.g., Antiquity or the Middle Ages), such cues are not commonly available, as no news articles from these eras exist. Generating event timelines for arbitrary historical periods is therefore a much harder problem, which requires methods that rely less heavily on the types of rich, parallel and dense information contained in news clusters.

To investigate this problem, we approach timeline generation as a special single-document summarisation task. In other words, we assume that the information to be summarised is contained in a single history article, and that no further mentions of specific events exist externally. This is a realistic scenario, for instance, for a specialist article describing the history of music in Ancient China.

We introduce a method for selecting salient content in history articles of any subject area, as long as the events in the text are roughly ordered chronologically. The hypothesis is that knowledge of an text's subject area can help decide which content should be selected. Another intuition is that certain combinations of events should be avoided in a timeline. We therefore investigate ways of encouraging a balanced selection of content from all parts of the text.

2 Related work

Timeline extraction has mostly been explored in a multi-document summarisation setting using corpora of news articles (Tran et al., 2015; Swan and Allan, 2000; Yan et al., 2011; Chieu and Lee, 2004;

Allan et al., 2001). This task definition allows the exploitation of features such as document creation times and headlines. The most important feature is redundancy between articles, which facilitates the identification of salient events.

A second important strand of work focuses on extracting *all* events from a single input text and anchoring them in time. The creation of the TimeML specification language (Pustejovsky et al., 2003) laid the foundations for the TempEval series of shared tasks (Verhagen et al., 2007; Verhagen et al., 2010; UzZaman et al., 2013), in which systems had to identify TimeML events and temporal expressions in free-form text. Further subtasks included the normalisation of temporal expressions and the creation of links between events and temporal expressions. A further shared task investigated the use of TimeML annotation for the downstream task of question answering (Llorens et al., 2015). Kolomiyets et al. (2012) created a connected timeline for a text based on TimeML annotations; a dependency parser infers dependency structures between events. Finally, a recent SemEval task (Minard et al., 2015) explored the related problem of cross-document event ordering. Here, relevant events and temporal expressions concerning a single target entity of interest have to be identified in more than one input document.

Chasin et al. (2014) try to identify important events in single texts, but their approach is limited to articles on wars and battles, and the problem is not approached as a summarisation task. Their method is lightly supervised, using features such as the presence of negation or past tense verbs in the sentence, and TextRank (Mihalcea and Tarau, 2004) for identifying salient sentences. We use an improved version of this system as a baseline.

3 Overall approach

Our problem is that of finding an optimal sequence of events (of a given maximum length) in a given input article. We follow the literature on event extraction and use TimeML events (Pustejovsky et al., 2003). Most TimeML events are verbs, but some are nominalisations such as "invasion" or other event-like words such as "war". The use of TimeML events, aside from the practical advantage that commonly-available event extraction algorithms exist, allows us to evaluate content selection at the event rather than at the sentence level.

We assume that there are both local and global factors that determine which events should be contained in the timeline. Local factors reflect how important an event is in its own right. Global factors represent intuitions about which combinations of events should or should not be selected. Our approach, which is unsupervised, takes into account the factors described in what follows.

3.1 Date presence

Intuitively, we expect that many important events have a date attached to them, as authors tend to give the reader this information if it is available. This is true for all historical periods from prehistory onwards, since for most events at least an approximate date is known. We considered two alternatives: The simplest approach is to only use sentences that contain a date, regardless of where in the sentence the date is located. A more sophisticated alternative verifies that the date is syntactically attached to the event, such as in "Richelieu died in 1642". To identify such cases, we constructed a parse tree using the C&C dependency parser (Clark and Curran, 2007) and only considered a TimeML event to be "dated" if it is at most two outgoing dependencies away from a temporal expression. We used HeidelTime (Strötgen and Gertz, 2013), a the state-of-the-art temporal expression software package, to identify such temporal expressions.

3.2 Lexical cues

The key component we use to judge the importance of any event are lexical cues about the input text's subject area. Examples of such subject areas include `INVENTION` and `FOOD/DRINK`. The subject area of a text should give us prior knowledge about which types of events are likely to be important. For instance, we would expect that a timeline describing the history of a country should contain information about revolutions, invasions, elections and similar events, whereas a timeline about science will instead focus on discoveries, publications, and Nobel prizes.

To mine knowledge about such subject-area-specific preferences, we make use of Wikipedia as a background corpus. Only history-specific articles whose title starts with "History of" are considered.

We start by generating sets of all Wikipedia history articles belonging to a given subject area, e.g. $A_{\texttt{GPE}}$ or $A_{\texttt{INVENTION}}$. To do this, we make use of the Wikipedia category system. For instance, for constructing a set of articles for the subject area `FIELD OF SCIENCE`, we collected all history articles that belong to the Wikipedia category "History of science by discipline". For each subject area g, we then calculate a *preference score* for each word lemma l found in any of the articles in the corresponding list A_g, using the following formula:

$$sc(g,l) = \frac{\frac{freq(A_g,l)}{freq(A_g,*)}}{\frac{freq(*,l)}{freq(*,*)}}$$

where $freq(A_g, l)$ is the summed frequency of word lemma l in all documents belonging to subject area g, and "*" stands for any. The numerator denotes how often lemma l appears in the subject-area-specific set of articles A_g, normalised by the total number of tokens found in this set. The denominator is invariant across all subject areas. If the ratio is high, lemma l is more likely to appear in articles of subject area g than in Wikipedia overall, suggesting that it is typical for the given subject area.

For each event e in the input text, a *local importance score* $imp(e)$ is calculated as

$$imp(e) = \frac{\sum_{w \in R(e)} \frac{sc(g,l)}{1+dist(w_e,w)}}{N}$$

where $R(e)$ is a window of words around the word representing the event (including the event word w_e itself), and $dist(w_1, w_2)$ refers to the absolute distance in words between two words w_1 and w_2. $imp(e)$ is a weighted average of the preference scores of all words in a window. The intuition is that context words of the event word can also be expected to be indicative of the subject area (consider "publish a paper") in many cases. $1 + dist(w_e, w)$ is used as a distance penalty in order to give more importance to words that are closer to the event word w_e. N is a constant which normalises the score by the sum of all distance penalties, to account for cases where the event word occurs at the beginning or end of a sentence. Table 1 shows examples of words with high and low preference scores.

3.3 Temporal coverage

We would like to avoid cases where too many events are selected from a small portion of the document,

GPE	INVENTION	FOOD/DRINK
absolutism	gas-works	yerba
protectorate	reverse-angle	hamburger
serfdom	flashback	saffron
club	season	play
game	team	member
season	school	bear

Table 1: Words with high (top) and low (bottom) preference scores for three subject areas

even if all these events are relevant. For instance, an article might list all a country's elections of the past few years, while mentioning only very important elections in earlier time periods. In this case, knowing that elections are important in the history of a country is not helpful, since this would lead to insufficient coverage of the remaining events in the article. We therefore take into account global factors as well. We experiment with two different methods:

Exploiting document structure. We select salient events from each section of the Wikipedia article in a round-robin fashion. The algorithm operates in a greedy fashion by selecting the most locally important remaining event for each section, until the desired timeline length has been reached.

Integer linear program. We use an integer linear program to encode the intuition that no two timeline entries should have the same year. The ILP maximises the following objective function for each article (E refers to the set of all dated events):

$$\sum_{e_i \in E} x_i \cdot imp(e_i) - \sum_{e_i \in E} \sum_{e_j \in E} b_{ij} \cdot pen(e_i, e_j)$$

subject to the constraints:

$$b_{ij} \leq x_i \; \forall i, j \in E$$
$$x_i + x_j - b_{ij} \leq 1 \; \forall i, j \in E$$
$$x_i \in \{0,1\} \; \forall i \in E \qquad b_{ij} \in \{0,1\} \; \forall i, j \in E$$
$$\sum_{e_i \in E} x_i = L_{max}$$

This is similar to the model used by McDonald (2007) for multi-document summarisation. The model tries to find a set of locally important events while discouraging the selection of events that have the same date. x_i is a variable denoting whether the corresponding event e_i has been selected. b_{ij} is a variable which is 1 if and only if both events i and j have been selected. $pen(e_i, e_j)$ is a penalty function that is 1 if the two events e_i and e_j have

the same date, otherwise 0. Each event was linked to the preceding temporal expression identified by HeidelTime; this heuristic was found to work well. The last constraint ensures that not more than L_{max} events are chosen, where L_{max} is the desired timeline length for the article considered.

4 Evaluation

For evaluating our algorithms, the methodology we introduced in (Bauer and Teufel, 2015) is used, along with the accompanying Cambridge Single-Document Timeline Corpus (CSDTC, version 2.0), which has been made publicly available[1].

4.1 Cambridge Single-Document Timeline Corpus

The CSDTC contains 10 articles from 3 subject areas: `GPE` (geo-political entities such as countries and cities), `FIELD OF SCIENCE` and `INVENTION`. To tune our algorithms, we constructed a development set of a further 30 annotated history articles from the subject areas in the CSDTC and one additional subject area (`FOOD/DRINK`). Due to the high annotation cost, only a single timeline creator was used. Important events were directly marked up in the source text (as opposed to the CSDTC, where timeline entries were written by hand), and exactly one HCU[2] was created per event. Using this development corpus, the window size of words considered for calculating local importance scores (cf. Section 3.2) was set to 3. We report the performance of all algorithms on both the development set and the test set (the CSDTC).

Although the number of subject areas in the two corpora is rather small owing to the considerable annotation effort, we believe that the resulting system would generalise rather well to other subject areas, were they added, as the subject areas in the corpus are very different in nature from each other. Care was taken when constructing the CSDTC to use a set of subject areas that is representative for human-written timelines on the Web.

[1]The corpus is available on the first author's website: `http://www.cl.cam.ac.uk/~smb89/form.html`

[2]As opposed to the CSDTC, HCUs in the development set always have a weight of 1, as only timeline writer was used.

4.2 Evaluation based on Historical Content Units

The evaluation is based on abstract ("deep") meaning units called *Historical Content Units* (HCUs). HCUs were derived on the basis of human-created timelines. Between 32 and 80 HCUs per article were annotated for the articles in the CSDTC.

Each HCU is weighted by the number of timeline creators who expressed its semantic content in their timelines. Because HCUs are linked to TimeML events in the surface text, it is possible to perform automatic deep evaluation without requiring any manual annotation of system summaries.

Algorithms are evaluated on a given input article using an adapted version of the pyramid score (Nenkova and Passonneau, 2004), which is calculated as the ratio between the sum of all rewards for HCUs chosen by the algorithm normalised by the maximum possible score $score_{max}$:

$$score = \frac{\sum_{h \in HCUs} w_h \cdot Cov(h,E,T)}{score_{max}}$$

where w_h is the weight of HCU h (a number between 1 and the number of annotators), E is the set of events in the article, T are the events in the system timeline, and the *coverage score* $Cov(h, E, T)$ is a number between 0 and 1 that indicates to what extent the events chosen by the algorithm jointly express the semantic content of HCU h. The basic version of $Cov(h, E, T)$ is defined as follows:

$$Cov(h, E, T) = min(1.0, \sum_{e_j \in E} v_{h,e_j} \cdot s(T, e_j))$$

where v_{h,e_j} is an *anchor weight* between 0 and 1 which denotes to what extent event e_j expresses the semantic content of HCU h, and $s(T, e)$ is a helper function that returns 1 if the set of selected events T includes event e, and 0 otherwise.

The coverage score for each HCU is calculated by summing up the anchor weights of those events that the algorithm has selected. A coverage score of 0 means that the events mentioned in the timeline do not express the HCU's semantic content at all, while a score of 1 occurs where the HCU's content is fully expressed by the timeline. Scores between 0 and 1 occur in a large number of cases. For instance, an HCU may express the fact that a country was invaded and destroyed. If the system timeline merely contains a TimeML event that refers to the invasion, it is assigned a coverage score of 0.5 for this HCU,

as it expresses only half of the HCU's semantic content. Where the sum exceeds 1, the coverage score is set to a hard upper limit of 1. This ensures that algorithms are not doubly rewarded for selecting multiple TimeML events expressing the same semantic content. The final formula we used to calculate coverage scores is slightly more complex, as some TimeML events in the CSDTC have been grouped together into *event groups*. A detailed description is given in the documentation of the corpus.

Pyramid scores are recall-based: The evaluation assumes a maximum number of timeline entries n, and the maximum possible score is the sum of the HCU weights of the n most highly weighted HCUs. The values for n are given in the CSDTC.

4.3 System and baselines

We report the performance of two systems. Both systems first remove all events that do not have a date, or whose date is too far away, as described in Section 3.1. Our first system ("ILP-based") selects events based on the integer linear program described, while the second system ("Round-robin") selects locally important events per section.

We have speculated above that dates are important for our task. We therefore compare against a date baseline which selects events randomly from the list of all dated events. We also compare against several modified versions of our method: To investigate the influence of the parser in identifying suitable dated events, we report the results for a simpler method which considers all events that have a date in the same sentence ("Round-robin, simple date criterion"). Two alternative systems select locally important events from all (not only dated) events ("Round-robin, without date criterion") or salient dated events from the entire article without considering document structure ("Local importance + date criterion").

The supervised baseline ("Chasin et al. (2014)") was re-implemented using LibSVM (Chang and Lin, 2011), and SVM parameters were tuned using grid search. 25 of the 29 articles were used for training and 4 for development. We improved their system by defining some of their sentence-level features at the event level. Probability estimates as described by Platt (2000) were used as importance scores.

System	Dev	Test
ILP-based	**0.22**▲	**0.30**▲
Round-robin	0.20▲	0.30▲
Round-robin w/o local importance	0.18	0.26
Local importance + date criterion	0.21▲	0.29
Round-robin, simple date criterion	0.19	0.25
Round-robin without date criterion	0.14	0.18
Date baseline	0.18	0.25
Chasin et al. (2014) (improved)	–	0.12
Random baseline	0.08	0.10

Table 2: Average pyramid scores across all articles (▲ = significantly better than the date baseline)

4.4 Results

The results in Table 2 show that only a combination of all three factors (date presence, local importance, coverage) results in a statistically significant improvement over the date baseline at $\alpha = 0.05$ according to Wilcoxon's signed-rank test on the test set. Both our systems perform comparably on the test set; removing any of the three components results in lower performance. Using a parser to identify dated events has a strong positive effect (see "Round-robin, simple date criterion"). Our system also outperforms the improved supervised baseline by a large margin. The fact that a completely unsupervised system performs best is encouraging, as training data for this task is very expensive to obtain. Our results suggest that it might be worth investigating other types of prior knowledge about the semantics of an input text in further research. The crucial advantage of such generic methods is that no texts on exactly the same topic are needed, which is a requirement with texts about niche topics.

5 Conclusion

We have introduced an unsupervised method for the challenging problem of timeline generation from single history articles, a scenario where parallel texts cannot be assumed to exist. Our method results in a significant improvement over a novel unsupervised baseline as well as an existing supervised approach.

Acknowledgments

The first author received financial support from Microsoft Research, St John's College Cambridge and the Cambridge University Computer Laboratory.

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 Information Retrieval*, SIGIR '01, pages 10–18, New York, NY, USA. ACM.

Sandro Bauer and Simone Teufel. 2015. A Methodology for Evaluating Timeline Generation Algorithms based on Deep Semantic Units. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 834–839.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Rachel Chasin, Daryl Woodward, Jeremy Witmer, and Jugal Kalita. 2014. Extracting and Displaying Temporal and Geospatial Entities from Articles on Historical Events. *Comput. J.*, 57(3):403–426.

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*, SIGIR '04, pages 425–432, New York, NY, USA. ACM.

Stephen Clark and James R. Curran. 2007. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics*, 33(4):493–552.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2012. Extracting Narrative Timelines as Temporal Dependency Structures. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 88–97.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015. SemEval-2015 Task 5: QA TempEval - Evaluating Temporal Information Understanding with Question Answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 792–800, Denver, Colorado, June. Association for Computational Linguistics.

Ryan McDonald. 2007. A Study of Global Inference Algorithms in Multi-document Summarization. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 557–564, Berlin, Heidelberg. Springer-Verlag.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Ruben Urizar. 2015. SemEval-2015 Task 4: TimeLine: Cross-Document Event Ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786, Denver, Colorado, June. Association for Computational Linguistics.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 145–152, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

J. Platt. 2000. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*.

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. In *Fifth International Workshop on Computational Semantics (IWCS-5)*, pages 1–11.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298.

Russell Swan and James Allan. 2000. TimeMine (Demonstration Session): Visualizing Automatically Constructed Timelines. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 393–, New York, NY, USA. ACM.

Giang Tran, Eelco Herder, and Katja Markert. 2015. Joint Graphical Models for Date Selection in Timeline Summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1598–1607, Beijing, China, July. Association for Computational Linguistics.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations.

In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 75–80, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 57–62, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary Timeline Summarization: A Balanced Optimization Framework via Iterative Substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 745–754, New York, NY, USA. ACM.