A Methodology for Evaluating Timeline Generation Algorithms based on Deep Semantic Units

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

Abstract

Timeline generation is a summarisation task which transforms a narrative, roughly chronological input text into a set of timestamped summary sentences, each expressing an atomic historical event. We present a methodology for evaluating systems which create such timelines, based on a novel corpus consisting of 36 human-created timelines. Our evaluation relies on deep semantic units which we call *historical content units*. An advantage of our approach is that it does not require human annotation of new system summaries.

1 Introduction

A timeline of historical events is a special kind of summary. We define a timeline as a list of textual event descriptions, each paired with a date (see Figure 1). A timeline is different from a standard single- or multi-document summary: Each event description is accompanied by a timestamp, and event descriptions themselves are independent linguistic units which should be understandable on their own. Additionally, a good timeline satisfies conflicting constraints: it should contain only salient events, and the overall time period considered should be covered well by events. Timeline construction is not a new task. It has been performed, for example, in a multi-document summarisation (Chieu and Lee, 2004; Yan et al., 2011; Nguyen et al., 2014) or in a single-document classification context (Chasin et al., 2013).

It is crucial to reliably evaluate algorithms that create such timelines automatically. Of course, any summary can be evaluated by surface methods such as ROUGE (Lin, 2004). But even for traditional summaries, ROUGE-based evaluation has been criticised for being too shallow, and it is even less adequate for timelines, because of their special properties described above.

> (...) In the 1997 unrest in Albania the general elections of June 1997 brought the Socialists and their allies to power. President Berisha resigned from his post, and Socialists elected Rexhep Meidani as president of Albania. Albanian Socialist Party Chairman Fatos Nano was elected Prime Minister, (...)

1997	There was unrest in Albania.
June 1997	Fatos Nano was elected Prime Minister.

Figure 1: Extract from a Wikipedia article and two lines of a corresponding timeline.

We therefore opt for a "deep" method which attempts to measure to which degree a system-generated timeline contains semantic units found in gold-standard timelines. Our content units resemble those of van Halteren and Teufel (2003) and Nenkova and Passonneau (2004), but are larger in that they correspond to historical events.

Traditional deep summarisation evaluation is expensive because it involves annotation of gold-standard summaries as well as annotation of each system summary. A major operational advantage of our approach is that we require human annotation only for gold-standard summaries, not for system summaries. After a one-time effort of creating semantic units and mapping them to the original text, the quality of a system's content selection can be evaluated for infinitely many new system summaries for free. Our method is the following:

1. Ask timeline writers to create timelines with a fixed number of date-event pairs.
2. An HCU creator (the first author) transforms these timelines into HCUs, historical content units, which are defined based on semantic overlap between timeline text.
3. We then create a mapping between HCUs and the source text, or more precisely, TimeML events in the source text. This mapping between HCUs and source text allows us to evaluate new systems without a human ever inspecting system output at all.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 834–839,
Beijing, China, July 26-31, 2015.

HCU 16	
Action	Fatos Nano is elected Prime Minister
Agent	*not given*
Patient	Fatos Nano
Time	June 1997
Location	Albania

Figure 2: HCU for one event from Figure 1

Any summarisation evaluation based on human judgment is inherently subjective, but we restrict this subjectivity in three ways. First, timeline creation (step 1) involves the selection of important content, which is by far the most subjective of the decisions involved in our evaluation method. We therefore ask three independent timeline writers to perform this task. Second, the generation of HCUs (step 2) is prescribed by fixed rules and definitions inspired by the methodology of van Halteren and Teufel (2003). With this method, the timeline writers, not the HCU creators, decide which material is available for creating the HCUs. Third, for the creation of mappings between HCUs and the source text (step 3), which was performed using a different set of detailed guidelines, we report agreement between the first and second author.

In section 2, we explain and contrast our concept of HCUs to existing work. In section 3, we present our new evaluation corpus and explain how we derived it. In section 4, we give details on how system scores for individual HCUs are calculated. In section 5, we analyse agreement of timeline writers on HCUs using two 3-person groups of annotators. We do not present our own algorithm for timeline generation here, but we sanity-check our evaluation methodology for a number of baseline timeline generation algorithms (section 6), where we demonstrate how systems are scored with our method.

2 Historical Content Units

Our event representation is called *Historical Content Unit* (HCU), which is inspired by the Summary Content Units (SCU) used in the pyramid method of Nenkova and Passonneau (2004) (henceforth NP04). Their approach is based on the idea that, due to the inherent subjectivity of summarisation tasks, there is no such thing as a single best gold standard summary. Instead, there are many equally good gold standard summaries. The way to differentiate between a good and a bad system summary is to consider each content unit selected by a system and count how many gold standard summaries it appears in. SCUs that are mentioned by many annotators contribute more to a system's score than less frequently chosen units. We follow this general weighting idea, but our HCUs are more abstract than SCUs, which are tied to a clause in the summary text without any further semantic characterisation by the annotator.

HCUs are more abstract in that they express an event, i.e. a concrete real-world action (*France invades Algeria*) or state change (*Obama becomes president*), while SCUs are more textual, not semantically defined and generally represent a smaller unit of meaning. State descriptions, opinions, wishes, aspirations, intentions and utterances do not constitute events. HCUs normally contain a logical agent (for actions) or a patient (for state changes), plus possibly other semantic roles. The action occurs at a given point in time, not as a continuous (e.g. "species adapt") or regular action ("the sun sets"), and the location of the event has to be delimitable, too (e.g., "in France" is acceptable, but not "on coral reefs"). An example HCU is given in Figure 2. Our HCU definition implies that each historical event is considered equally important. For system evaluation, this means that a system can score at most one point per HCU (exactly one point if it gives a perfect rendition of that HCU). This is different to evaluation based on the SCUs in NP04. Their method of linking words to SCUs may lead to a situation where some events are represented by multiple SCUs, and hence are effectively considered more important than others.

HCU construction proceeds by treating each line in each timeline as a single HCU candidate. If a line contains more than one event (for instance an event plus additional information), we decide what the main event is based on syntactic criteria and discard the additional information. We then have to decide whether two or more surface string descriptions of events by different timeline authors correspond to the same HCU. For this, we follow the method described by van Halteren and Teufel (2003). As long as two event descriptions do not contain conflicting information about an event and as long as their timestamps do not disagree, we can safely assume they refer to the same real-world action and map them to the same HCU, for instance, the two sentences "Nano was elected Prime Minister" and "Party Chairman Nano was elected PM".

This matching process results in a number of

HCUs for a source text, each with associated surface representations by human timeline authors. In the future, these gold standard event realisations could be used to evaluate the surface form of system timelines. This paper, however, is mainly concerned with content selection evaluation.

We now use the number of surface representations available to assign a weight to each HCU (following NP04), and the following formula is used to calculate the total score for a system:

$$score = \frac{\sum_{i \in HCUs} w_i \cdot score_i}{score_{max}}$$

where $score_i$ denotes the individual scores (between 0 and 1) calculated for each HCU i and w_i is the number of annotators whose timeline contains that HCU. $score_{max}$ is the sum of the weighted maximum scores of the n most highly weighted HCUs in the pyramid, where n is the desired timeline length.

3 Corpus construction

To find suitable historical articles for our corpus, we created the intersection of all Wikipedia articles whose title starts with "History of" with the articles in a large collection of timelines described by Bauer et al. (2014). Articles with errors in their Wikitext were removed. We also excluded articles that were incomplete, did not contain narrative text or were not chronologically structured. None of these criteria aim at hand-picking well-written articles or articles that describe well-attested topics. The final set consists of 408 articles. We manually grouped these according to their general topic area (`GEO-POLITICAL ENTITY`, `SCIENCE`, ...). From these, we select a set of 11 articles representative in terms of length and subject area. For each of the articles in our corpus, we removed the introductions (which tend to contain a summary of the entire article). We then asked 3 annotators per text[1] to produce a historical digest with a given maximum length determined by the number of verbs in each article (resulting in 25-40 events). For one text, we asked an additional 3 annotators to provide timelines, such that this one text was covered by six annotators. This means that in total, we had 36 combinations of texts and timeline writers.

Our instructions do not tell the timeline writers how content selection (in the source text) and surface realisation (in the timelines produced) should be performed. We merely state that the timeline should strike a balance between mentioning all and only important events and still giving a complete account of the time period covered. Annotators are also told that each line should contain exactly one event and must be given a timestamp.

Our approach brings with it the challenge of deciding when an algorithm operating on the source text has correctly selected an HCU. We assume that individual words in the text – verbs, nominalisations and certain other event-like nouns (such as "war") – are associated with the core action or state change expressed by an HCU and that we or a system can find those. While our methodology does not presuppose any particular event definition or event extraction paradigm, we make use of the TimeML project (Pustejovsky et al., 2003), which has provided a substantial body of work on how to extract events and timestamps in the form of TimeML `EVENT` and `TIMEX` instances (cf. the TempEval shared tasks (Verhagen et al., 2007; Verhagen et al., 2010; UzZaman et al., 2013)).

To construct the links between the HCUs we found in the texts (between 32 and 80 per text[2]) and the surface text, we first run a publicly available, recent TimeML-based extraction system, TIPSem-B (Llorens et al., 2010), over our texts. We then manually annotate each HCU with all surface sentences that express the action or state change described by the HCU, and manually decide which TimeML event(s) identified in any such matching sentence express(es) the HCU's content. This results in a 1:n mapping between HCUs and events. For this matching process, we use a detailed set of guidelines. A subset of the 2066 matchings (all matchings for 60 HCUs) was re-annotated independently by the second author; inter-annotator agreement was 87.9%.

Where the TimeML system failed to recognise what we consider to be the correct event anchor, we manually tagged this event anchor, and we provide this information with our corpus. This is because we want our gold standard to be independent of any particular event extraction package.

4 Scoring system

As stated above, the reward a system may receive for a single HCU is capped to one. This is true regardless of how many TimeML events represent-

[1] The annotators recruited included both computational linguists and students in higher education.

[2] We obtain 100 HCUs for the text annotated by 6 humans.

ing the HCU are retrieved by the system. We uphold this principle because we aim to evaluate how many HCUs a system returns, not how many textual elements representing them are retrieved.

Apart from this global constraint, the general principle is to treat the contribution of each individual TimeML event additively. For example, if three events have been found to represent a third of the meaning of the HCU, respectively, and two of them are selected by the system, the total score obtained for this HCU will be $\frac{2}{3}$.

For some pairs of TimeML events, however, this additive paradigm is not the desired behaviour: The TimeML software sometimes tends to mark two very closely related words, e.g. a verb ("start") and its object ("war"), as events. In this case, we do not want these two events, which we consider to be members of an *event group*, to contribute additively (AND); instead, an OR logic is appropriate, meaning that it is irrelevant whether one or both of the participating events are chosen. The human matcher may impose such constraints between multiple events linked to the same HCU.

In general, an event group E is a set which may contain individual events $e_1, e_2, ...$ and further subgroups $E_1, E_2, ...$ of events.

$$E = \{E_1, E_2, ..., E_n, e_1, e_2, ..., e_n\}$$

Each event and subgroup in E is associated with a number $v \in [0, 1]$ that denotes how much the event or subgroup contributes to the total meaning of event group E in context of HCU i; these numbers are set by the human matcher.

The total $score_i$ that a system will receive for an HCU i is calculated using the function $S(i, E)$, where E is an event group that includes all events linked to HCU i by a human:

$$S(i, E) = min(1, \sum_{E_j \in E} v_{E_j} \cdot S(i, E_j) + \sum_{e_j \in E} v_{e_j} \cdot s(i, e_j))$$

$S(i, E_j)$ represents the contribution made by all TimeML events in an event subgroup $E_j \in E$, which is again capped to 1 via the recursive definition of the score function. $s(i, e)$ is a function for an individual event in group E which, if the system to be evaluated has chosen the event, returns one, and zero otherwise:

$$s(i, e) = \begin{cases} 1 & \text{if the system has chosen event } e \\ 0 & \text{otherwise} \end{cases}$$

Note that $S(i, E)$ simplifies to $s(i, e)$ if there is only one event e linked to HCU i (and if $v_e = 1$).

See Figure 3 for an example of an HCU along with all TimeML events in a sentence from the source article and their respective contributions to the HCU's meaning (in brackets). Here, the matcher has decided, according to our guidelines, that "began" fully represents the HCU's meaning, while "recording" only represents half of the meaning. Importantly, a system selecting both these events will still only receive a total score of 1.0 for this HCU since it is capped to that number.

5 Data analysis

While we do not expect perfect agreement for timeline generation, we hope to observe a pyramid form like in NP04; i.e. a situation where few HCUs are chosen by all three annotators, a higher number are chosen by two annotators, and so on. Indeed this was the case for 9 of the documents.

We also investigated how different the gold standard would have been if a different set of three humans had annotated the texts. We asked three further annotators to create historical digests of one text and then considered all possible splits into two groups of three annotators each. For illustration, Tables 1 and 2 represent two examples out of the 10 possible configurations, showing the number of annotators per group that agreed on HCUs. The grey areas in the tables capture cases where the two annotator teams chose an HCU with the same frequency or where the two frequencies differ only by one. Averaged across the 10 splits, 91.9% of all HCUs fall into this area.

Consider cell (#0, #0): These are the cases where all six annotators decided that these events are not worthy of being mentioned in the timeline. Since we do not annotate non-selected HCUs, we can only give an approximation for this number based on the average observed HCU frequency per sentence. We do this since these cases should arguably also contribute posivitely to the agreement. Using these tables, we calculate Krippendorff's α across annotator groups; i.e. each HCU can receive a score between 0 and 3, depending on how many annotators expressed it in their timeline. We use an interval difference function and obtain $\alpha = 0.530$. This is arguably a non-standard use of α; we provide this number to give the reader a rough idea of the agreement across groups.

6 Baseline results

To illustrate our method, we now present the results of a number of baseline algorithms. We only evaluate the systems' choice of events, not the sur-

Action	The Southern Semites began recording their history
Agent	the Southern Semites
Patient	their history
Time	800 BC
Surface text	This **led (0.0)** to **contact (0.0)** with the Phoenicians and from them , the Southern Semites **adopted (0.0)** their writing script in 800 BCE and **began (1.0) recording (0.5)** their history .

Figure 3: Example HCU with links into the surface text (the HCU's location is not given)

		Team 2			
		#0	**#1**	**#2**	**#3**
Team 1	**#0**	87	19	2	1
	#1	18	15	10	1
	#2	6	8	9	4
	#3	1	0	3	3

Table 1: Best split (94.1% in grey area)

		Team 2			
		#0	**#1**	**#2**	**#3**
Team 1	**#0**	87	17	6	0
	#1	20	9	6	3
	#2	8	14	5	1
	#3	0	2	6	3

Table 2: Worst split (89.8% in grey area)

face realisation or the timestamps. In the future, a more sophisticated mechanism may be devised which takes these aspects into account as well.

Our algorithms are listed in Table 3. They may select individual TimeML events from the source text (1-6), or entire sentences (7-10); in the latter case, all events in the sentences count as selected. Some of the baselines select events from anywhere in the article (4, 5, 6, 10); others proceed in a round-robin fashion by iteratively selecting one event or sentence per section (1, 1b, 2, 3, 7, 8, 9, "RR"). For the latter methods, in each iteration we can proceed from the top (1, 1b, 7) or the bottom (2, 8) of the section, or we randomly select any event or sentence in the section (3, 9). Choosing the first or the last events of the entire article (5, 6) does not look like a good method, since the timeline needs to cover the entire timespan. Finally, we examine whether selecting only events with a date in the same sentence has any effect; results can only be calculated over 10 articles since one of the articles does not contain enough such events. The result in Table 3 is therefore marked with a star (*). The results of methods that involve randomly selecting items were averaged over 100 runs. In principle, existing systems such as that by Chasin et al. (2013) could also be evaluated with our method, but we do not do this here.

ID	Method	Scores
1	RR, first events in section	0.23
1b	like 1, events with dates only	0.33*
2	RR, last events in section	0.13
3	RR, random events in section	0.13
4	Random events in article	0.11
5	First events in article	0.13
6	Last events in article	0.11
7	RR, first sentences in section	0.22
8	RR, last sentences in section	0.11
9	RR, random sentences in section	0.12
10	Random sentences in article	0.10

Table 3: Baseline results (average pyramid scores); the result with a * is based on 10 articles

Algorithms inspired by the well-established "first n words" baseline for summarisation of newswire articles perform best here too, when applied on a section level (1, 1b, 7). All these algorithms perform significantly better when compared to any of the other algorithms (2-6, 8-10); statistical significance is measured for each pair of algorithms at $\alpha = 0.05$ using the Wilcoxon signed-rank test ($p < 0.05$). This suggests that important events tend to be placed at the beginning of a section. Selecting the first events from the entire article (5) produces worse results than selecting the first events from each section. The best results are obtained when selecting only events with dates in their proximity (1b); however, this result is based only on 10 of the 11 articles, and the difference to algorithm 1 is not significant ($p = 0.1391$).

7 Conclusion

We have introduced a novel methodology for evaluating timeline generation algorithms based on deep semantic content units, including a new corpus of 36 human-written timelines and associated HCUs. Our evaluation focuses on a deeper model of meaning (based on events) rather than n-gram overlap, and provides links between each HCU and the source text. This allows us to subsequently evaluate an unlimited number of system summaries without any further cost, rationalising the evaluation of timeline construction algorithms.

References

Sandro Bauer, Stephen Clark, and Thore Graepel. 2014. Learning to Identify Historical Figures for Timeline Creation from Wikipedia Articles. In Luca Maria Aiello and Daniel A. McFarland, editors, *Social Informatics - SocInfo 2014 International Workshops, Barcelona, Spain, November 11, 2014, Revised Selected Papers*, volume 8852 of *Lecture Notes in Computer Science*, pages 234–243. Springer.

Rachel Chasin, Daryl Woodward, Jeremy Witmer, and Jugal Kalita. 2013. Extracting and Displaying Temporal and Geospatial Entities from Articles on Historical Events. *The Computer Journal*, pages 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, Sheffield, United Kingdom.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Hector Llorens, Estela Saquete, and Borja Navarro. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 284–291, Los Angeles, California. 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.

Kiem-Hieu Nguyen, Xavier Tannier, and Véronique Moriceau. 2014. Ranking Multidocument Event Descriptions for Building Thematic Timelines. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1208–1217, Dublin, Ireland.

James Pustejovsky, José M. Castaño, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. In Mark T. Maybury, editor, *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA*, pages 28–34. AAAI Press.

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.

Hans van Halteren and Simone Teufel. 2003. Examining the Consensus Between Human Summaries: Initial Experiments with Factoid Analysis. In *Proceedings of the HLT-NAACL 03 on Text Summarization Workshop - Volume 5*, HLT-NAACL-DUC '03, pages 57–64, Edmonton, Canada. 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, Prague, Czech Republic. Association for Computational Linguistics.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden, July. 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, Beijing, China.