This is a previous version of the article published in Information Processing & Management. 2019, 56(5): 1775-1793. doi:10.1016/j.ipm.2019.02.010

NATSUM: Narrative abstractive summarization through cross-document timeline generation

Cristina Barros, Elena Lloret, Estela Saquete, Borja Navarro-Colorado

Department of Software and Computing Systems, University of Alicante
Apdo. de Correos 99 E-03080, Alicante, Spain
{cbarros,elloret,stela,borja}@dlsi.ua.es

Abstract

A new approach to narrative abstractive summarization (NATSUM) is presented in this paper. NATSUM is centered on generating a narrative chronologically ordered summary about a target entity from several news documents related to the same topic. To achieve this, first, our system creates a cross-document timeline where a time point contains all the event mentions that refer to the same event. This timeline is enriched with all the arguments of the events that are extracted from different documents. Secondly, using natural language generation techniques, one sentence for each event is produced using the arguments involved in the event. Specifically, a hybrid surface realization approach is used, based on over-generation and ranking techniques. The evaluation demonstrates that NATSUM performed better than extractive summarization approaches and competitive abstractive baselines, improving the F1-measure at least by 50%, when a real scenario is simulated.

Keywords: Narrative summarization, Abstractive summarization, Timeline Generation, Temporal Information Processing, Natural Language Generation

1. Introduction

Managing and processing the over-abundance of information and its heterogeneity is an enormous challenge for human beings in the digital era. Therefore, the application of Human Language Technologies (HLT) is necessary to facilitate access to and use of this information. For example, every day, online newspapers generate countless digital texts (news) about the same facts. In this context, a summary is useful to support humans in the analysis and processing of information [1]. Text summarization can provide appropriate mechanisms to automatically condense the key information that is spread over different documents (e.g. news) [2].

To provide users with easy and optimal access to all this information, summaries must provide a coherent and natural structure. In this sense, narrative structure is the most natural and friendly text structure for human beings [3]. As human beings, we tend to organize the flux of happening in narrative structures, where a narrative structure is the arrangement of a set of events about one or more entities following a time order (that could be natural chronological order —from past to future— or artificial order —with time jumps—). Each event is a fact that occurs in the (real or imaginary) world at a specific moment with a specific structure (the event structure) [4], and denotes processes, activities, states, achievements or accomplishments [5]. Furthermore, an event involves participants [6] and other components that complete the event such as time, place, instruments, patients, etc[1].

Depending on how a summary is produced, a distinction can be made between *extractive* and *abstractive* summaries. *Extractive* summaries are produced by directly selecting the most significant sentences of a document and copying them verbatim into the output. *Abstractive* summaries are more challenging, since they include new or different vocabulary, linguistic expressions or concepts that do not originally appear in the input documents, but that paraphrase the most relevant information of the input. When the summary is intended to narrate or describe a series of events that happened at a specific time, *extractive* summarization approaches will lose the temporal connections appearing in the text, that can lead to dangling references,

[1]From a linguistic point of view, the participants and components of an event are called "arguments" and "modifiers". An event mention is formed by an event head (normally a verb, but not always), a set of arguments and optional complements. The arguments are those elements of the event structure that complete the meaning of the verb (as, for example, the person that carries out the specific action expressed by the verb, the person or object that receives the action, the instrument used to perform the action, etc.). The modifiers are the remaining optional elements of the event structure (the place where the action occurs, the time, etc.). In this paper, the word "argument" is used as a linguistic term to refer to the elements of the event structure [4]. Given that there is no common typology of arguments in the linguistic literature, we follow the proposal of PropBank project [7] to nominate arguments with numbers from A0 (the argument closest to the verb) to A4 (the most external argument), and AM for the remaining modifiers.

and thus the resulting text may be ambiguous or difficult to understand. For instance, an *extractive* summarization system could select the sentence *"Terrorists provoked the blast"* from the text shown in Example 1 without providing any additional information about other relevant information, such as *when?* or *where?*. However, using an *abstractive* summarization approach, the relevant information (e.g., *who? what?, when?, where?,...*) could be fused together, leading to the generation of one or more new sentences. Following the same text fragment given as example (Example 1), the sentence *"On Friday, terrorists exploded bombs in the U.S embassy in the Kenyan and Tanzanian capitals."* could be generated.

(1) Suspected bombs [exploded $_{event}$] outside the U.S. embassies in the Kenyan and Tanzanian capitals [Friday $_{time}$]. Terrorists provoked the [blast $_{event}$]

However, although *abstractive* summarization would be more appropriate than *extractive* summarization, the detection and resolution of temporal information is of crucial importance to anchor the event to a precise date. This avoids reader misunderstanding, (e.g. instead of *"On Friday"*, it would be more appropriate for ordering purposes to reformulate the expression as *"On the 7th of August 1998"*). In this way, the final summary would be clearer, containing all the relevant information within a coherent and cohesive text, thereby removing any possible ambiguity.

The main objective of this paper is to develop an abstractive summarization approach that generates narrative summaries based on a natural time ordering of events from a set of documents (news in this case) that deal with the same real events. Hereafter we will refer to it as the acronym NATSUM (Narrative Abstractive Timeline Summarization). This system has two main components: (i) a cross-document timeline generation module that extracts events related to the same entity from several texts (cross-document) and the time slot in which each event occurs, arranging them in a timeline; and (ii) an abstractive summarization module that transforms these time-ordered events into a single text with a time-based chronological narrative structure.

The task of extracting events involving a particular target entity among different documents and ordering them chronologically is known as Cross-document Timeline Extraction [8]. Timeline Extraction comprises the accomplishment of three stages. The first step involves determining whether the events extracted from the different documents are related to the target topic or entity. From this first cluster of events, a *temporal information processing* is required in order to extract the temporal expressions and the temporal relationships established between these events, determining thus which events happened at the same time. Finally, *cross-document event coreference* is needed in order to cluster all the mentions that occur at the same time and actually refer to the same event, regardless of the words used to express them. The previous Example 1 contains two event mentions[2] that refer to the same event.

For the creation of the narrative abstractive summary, a single sentence for all the events mentions referring to the same event is generated. This sentence includes all the information related to this event as well as the time it occurred. In this way, the abstractive summaries will be generated over the structured knowledge previously obtained from an enriched timeline[3]. This implies an advance on classical timeline extraction as it involves the addition of all the arguments related to the event. Also, there is an improvement in automatic narrative summarization as the temporal information (temporal expressions, events and temporal relationships) is considered in the summary generation process.

The paper is organized as follows. Section 2 contains a detailed background study of the different relevant research fields, involving Automatic Timeline Generation, Abstractive Summarization and Natural Language Generation. Section 3 describes the architecture of our proposed system NATSUM. Following this, Section 4 presents the main experiments conducted together with the evaluation methodology. Section 5 reports on the results obtained and a discussion of the findings. Furthermore, Section 6 reports additional experiments and evaluation to assess NATSUM's performance within the similar task of timeline summarization and compare its results to the state of the art. Finally, Section 7 highlights the main conclusions of this research and outlines some potential areas of future work.

2. Background

Considering that our proposal is generating narrative abstractive summaries based on timeline knowledge, both research issues are tackled in this section.

---

[2]Event mention is a reference to an event, that is, the different forms to refer to the same event.

[3]We propose summarization focused on a target entity because we are using the timelines defined in Semeval2015 Task 4, which defined timelines related to a target entity.

2.1. Automatic Timelines

Recently, SemEval-2015 [9] included a task that tried to combine temporal information processing and event coreference to obtain a timeline of events related to a specific given entity, from a set of documents [8]. They proposed two different tracks on the basis of the data used as input. Track A, for which they provided only raw text sources, and Track B, for which they also made gold event mentions available.

Track A had two participants: WHUNLP team, that processed the texts with Stanford CoreNLP[4] [10] and applied a rule-based approach to extract target entities and their predicates and also performs temporal reasoning[5] and the SPINOZAVU [11] system, that is based on a pipeline, developed in the NewsReader project, and addressed entity resolution, event detection, event-participant linking, coreference resolution, factuality profiling and temporal relation processing, first at document level, and then at cross-document level, in order to obtain timelines.

Track B had also two participants: Heideltoul team approach [12] that uses the HeidelTime tool for temporal information processing, and the Standord CoreNLP for event coreference resolution. A cosine similarity matching function and a distance measure are used to select which sentences and events are relevant for the target entity. Finally, GPLSIUA team [13], that uses the OPENER language analysis toolchain[6] for entity detection, the TIPSem tool [14, 15] for temporal processing and a topic modeling algorithm over WikiNews corpus to detect event coreference.

Outside SemEval-2015 competition, the work presented by Laparra et al. 2017 [16] developed three deterministic algorithms for timeline extraction based on two main ideas: a) addressing implicit temporal relations at document level, and b) leveraging several multilingual resources to obtain a single, interoperable, semantic representation of events across documents and across languages.

The novelty of our proposal is going further with the timeline extraction task, including all the participants in the events, and combining this technique with a summarization approach to generate narrative and ordered texts related to a specific topic.

[4] http://stanfordnlp.github.io/CoreNLP/

[5] No bibliography is available apart from the general paper of SemEval 2015 Task 4

[6] http://www.opener-project.eu/webservices

2.2. Abstractive Summarization and Natural Language Generation

As it was stated in the previous section, abstractive summarization is far more challenging than extractive summarization, since it requires understanding the information expressed in one or several documents and compress, fuse, integrate, enrich or generalize it to create a new text (i.e., summary) that contains the key aspects of the input documents. For generating high quality abstractive summaries, the integration of Natural Language Generation (NLG) techniques are crucial to be able to paraphrase the information expressed in the original sentences.

NLG tasks are commonly viewed as a pipeline of three broad stages: document planning (also known as macroplanning), microplanning and surface realization [17]. In the document planning stage, the system must decide what information should be included in the text and how to organize it into a coherent structure, leading to a document/text plan. From this document plan, in the microplanning stage, a discourse plan will be generated, where appropriate words and references will be brought together into sentences. Finally, the surface realization stage generates the final text with the information and structure selected. Each of the stages described has different goals and tasks to complete. In some research they are dealt with one at a time, or they focus on one task in particular. As examples of the latter, some popular tools developed in the context of NLG include SimpleNLG [18], which prioritizes the realization stage, or more specialized tools such as AIGRE [19], whose focus lies on the referring expression generation task. There have been some attempts to address the whole process as well, mostly using machine learning techniques. For instance, Duma and Klein [20] proposed that automatic template acquisition, and learning the content selection, output structure and the lexical choices to display take place simultaneously in a single process. Konstas and Lapata [21] analyzed several mechanisms for mapping database information (weather forecast records) into natural language sentences. These included the use of probabilistic grammars, the detection of patterns in input records and the learning of rhetorical relations to provide document plans from these records.

As regards the techniques used for automatic language generation, since this is not a trivial task, NLG systems have used either statistical or knowledge-based approaches. The underlying idea of statistical approaches is based on the probability of certain words appearing together and/or in proximity, studying the creation of a sentence on the basis of a set of words [22, 23]. In contrast, knowledge-based approaches use linguistic theories, e.g., rhetorical structure theory, to generate the text [24]. The fundamental difference between these approaches is the type of data used. Knowledge-based approaches use linguistic information (morphological, lexical, syntactic, semantic), together with rules and pre-defined templates. Statistical approaches use probabilistic information extracted from a text corpus. It is also important to note that rule-based knowledge approaches are oriented to a specific domain and language. Consequently, their adaptation to a different domain or language is extremely difficult and costly. In this sense, statistical approaches offer an advantage, since they are more versatile for application across different domains or languages, as long as the probabilities are learned from the appropriate corpora. Languages models (LM) can be considered one of the most-used mechanisms from the statistical perspective in HLT [25]. To obtain knowledge from a corpus on frequency and probability of word appearance — the fundamental idea behind LMs — several techniques can be applied: maximum likelihood [26] and support vector machines [27] have been widely used, for example.

In contrast to the NLG techniques for tackling abstractive summarization, other techniques employing neural networks models have emerged in recent years. For instance, See et al. [28] present a hybrid pointer-generator architecture with coverage for multi-sentence abstractive summarization. Chen and Bansal [29] propose a fast summarization model that generates a concise overall summary by selecting and rewriting salient sentences abstractively. These types of models tend to contain redundant and/or repeated information in the summary. In addition to these techniques, there are others that, in some way are a middle-ground between abstractive and extractive techniques. Examples of these types of techniques can be found in Cordeiro et al. [30] where a methodology for learning sentence reduction is presented; or in Valizadeh and Brazdil [31], where a summary is generated by selecting the sentences which satisfy actor-object relationships.

Our summarization approach is completely abstractive, focusing only on the surface realization stage, since the cross-document timeline generation will be used as a document plan. Moreover, different from the state of art, to generate a sentence, our approach will combine a statistical model together with semantic information, thus resulting in an hybrid surface realization method.

2.3. Narrative structures extraction

To the best of our knowledge, we are not aware of any previous work that attempts to generate narrative abstractive summaries using timeline information and NLG techniques. However, some previous proposals exist that attempt to extract event-based narrative structures from texts. Chambers and Juravsky [32, 33] extract narrative chains that define a partially ordered sets of events that share a common actor (an entity person). The relationship between events is, in this case, time relations. Our approach is based on these narrative chains. Similar approaches are used by [34], [35] or [36] to create narrative chains, but their work is focused on the extraction of common sense knowledge for a complete understanding of narrative texts. All these proposals extract the narrative chains from only one text. Our approach is, however, cross-document. We extract a single timeline of events (as a narrative chain) from several texts that talk about the same entity and about the same events.

Regarding timelines, a task close to our proposal is timeline summarization. According to [37], given a query (such as "BP oil spil"), timeline summarization 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 ([38] [39] [40] [41] [42] [43] [44]). Formally, a timeline is a sequence $(d_1, s_1), \ldots ,(d_k, s_k)$ where the d_i are dates and the s_i are summaries for the dates d_i, given a query q and an associated corpus C_q that contains documents relevant to the query. The task of timeline summarization is to generate a timeline s_q based on the documents in C_q. The number of dates in the generated timeline, as well as the length of the daily summaries, are typically controlled by the user. However, the aim of our proposal is to generate narrative summaries and not timelines, whereby timelines are used to generate the narrative structure, which means that the input of the summarization module is a target oriented timeline and not a set of documents, as in TS approaches.

The next section presents how the summary generation is performed, based on the arrangement of events along a timeline.

3. Narrative Abstractive Timeline Summarization System (NATSUM): Design and Development

The task we address consists of producing an abstractive multi-document summary that narrates the most relevant events[7] together with the date they occurred and when a specific target entity is involved. In this way, as shown in Figure 1, given as an input a target entity and a set of documents related to that target, the proposed system has to i) determine which events happened and when, choosing only the most relevant ones related to the target entity, building a timeline, which is used to ii) generate the final abstractive summary as output.

Therefore, the architecture of NATSUM comprises two different modules and it uses a set of news documents and a target entity as input. The two modules of the architecture are as follows:

- Enriched Timeline extraction: This module structures all the information related to a specific topic/target entity in a timeline. All the event mentions happening at the same time and referring to the same event are grouped together on the timeline. This module is an improved extension of the system presented in [45] .

- Abstractive summarization: This module is responsible for generating a chronological abstractive summary based on NLG techniques given an enriched timeline as input. Specifically, it employs a hybrid surface realization approach, based on over-generation and ranking techniques.

The integration of both modules as a pipeline results in the generation of a narrative abstractive summary. The proposed architecture is graphically depicted in Figure 2. In the following sections, the development of each of the aforementioned modules is explained in more detail.

3.1. Enriched Timeline extraction

As previously explained, given a set of documents and a set of target entities, the original task of Cross-Document Timeline Extraction consists of building an event timeline for a target entity from a set of documents [46].

[7] According to TimeML temporal annotation schema "events" is something that happens or occurs. Events can be punctual or last for a period of time. They also consider as events those predicates describing states or circumstances in which something obtains or holds true.

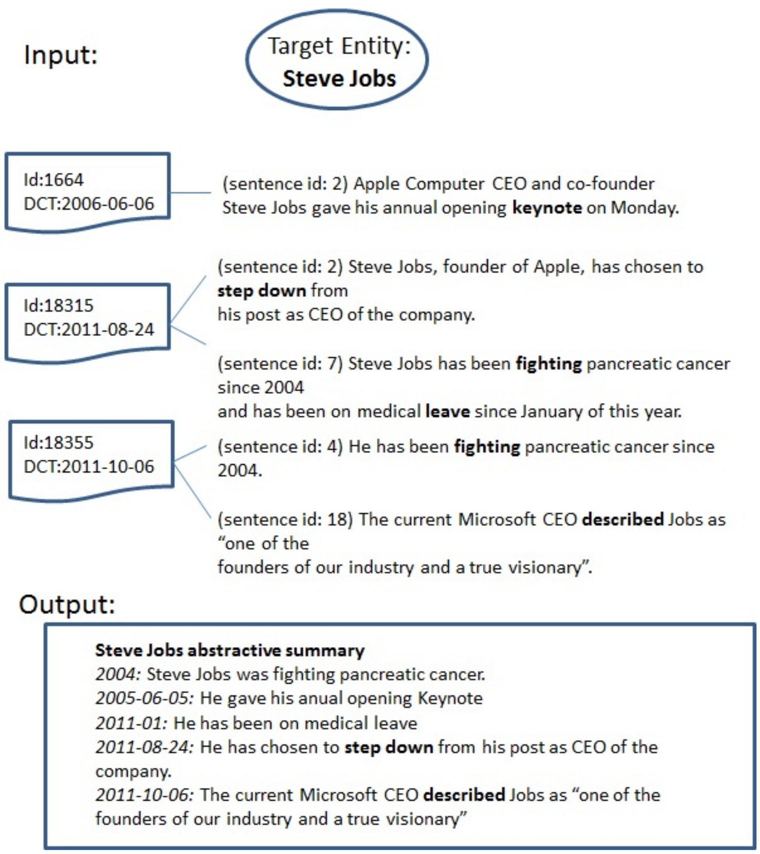

Figure 1: Example of input/output of the proposed system (NATSUM)

Theoretically, the main idea of our approach is that two events *e1* and *e2* will be coreferent if they are not only temporal compatible $(e1_t = e2_t)$[8] but also if they refer to the same facts (semantic compatibility: $e1_s \simeq e2_s$)[9]:

$$\text{coref(e1, e2)} \rightarrow (e1_t = e2_t) \wedge (e1_s \simeq e2_s) \quad (1)$$

Our proposal extends the approach by enriching the event clusters with

[8] ei_t: Temporal information of the event i
[9] ei_s: Semantic information of the event i

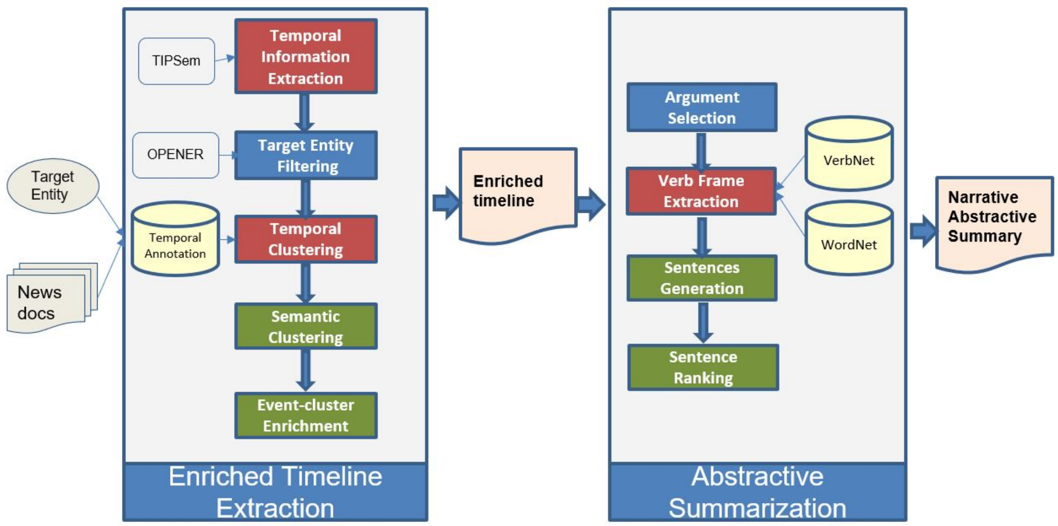

Figure 2: Architecture for our Narrative Abstractive Timeline Summarization system (NATSUM).

all the arguments extracted from these events in the different documents where they are presented. The steps of this module are:

- Temporal clustering[10]: by using the temporal information annotated by a temporal information processing system, the temporal relations between the events are processed and the events can be ordered and anchored to the timeline.

- Semantic clustering: the events are grouped together using event type information and distributional semantic knowledge.

- Event cluster enrichment: for each cluster of events, all the arguments related to the events in the cluster are added to the cluster.

3.1.1. Temporal Information Extraction

The input is a set of plain texts, and, therefore, the events in those texts must be automatically extracted. Furthermore, considering that the

[10]Temporal clustering in this context refers to Temporal Compatible Grouping, meaning that all the events happening at the same time are grouped together in a cluster. It is not the same concept as clustering in Machine Learning

final aim is building a timeline, temporal expressions and temporal links between events and times are required. Therefore, plain texts need to be annotated with all the temporal information. Several efforts have been made to define standard ways to represent temporal information in texts. The main objective of this representation is to make temporal information explicit through standard annotation schemes. TimeML[47] is the most standardized schema and it annotates not only events and temporal expressions, but also temporal relations, known as links [48]. In this annotation schema, event is used as a cover term to identify *something that can be said to obtain or hold true, to happen or to occur*. This notion can also be referred to as eventuality including all types of actions (punctuals or duratives) and states as well (section 1, NewsReader Guidelines[11]). Besides, according the task definition of Semeval 2015 —task 4, not all events can be part of a TimeLine, amongst others, counter-factual events will not appear in a TimeLine. Example (2) shows a sentence annotated with TimeML temporal expressions (`TIMEX3`), events (`EVENT`), and the links between then(`TLINK`).

(2) John `<EVENT eid="e1">came</EVENT>` on `<TIMEX3 tid="t1">Monday</TIMEX3>`
`<TLINK eventInstanceID="e1" relatedToTime="t1" relType="IS_INCLUDED" />`

In our case, the first step is performing Temporal Information Extraction and Processing, and TIPSem system (Temporal Information Processing using Semantics) [14, 15][12] is used for this purpose. TIPSem is able to automatically annotate all the temporal information according to TimeML standard annotation scheme [47], which means annotating all the temporal expressions (TIMEX3), events (EVENT) and links (TLINKS) between them.

3.1.2. Target Entity Filtering

Considering that not all the events are necessary to build the timeline, but only the ones related to a target entity, a Target Entity Filtering needs to be performed in order to discard those events that are annotated but not related to the given entity. The Target Entity Filtering requires resolving name entity recognition and entity coreference resolution, and OPENER[13] web services are used for this purpose. To determine whether an event should be part of the timeline, this module chooses: a) the events in which a target entity (or a

[11]http://www.newsreader-project.eu/files/2013/01/NWR-2014-2.pdf
[12]`http://gplsi.dlsi.ua.es/demos/TIMEE/`
[13]http://www.opener-project.eu/webservices

target entity coreference) explicitly participates in a *has_participant* relation with the semantic role A0 (i.e. agent) or A1 (i.e. patient), as defined in the Propbank Project [7], and b) in case of nominal events, since the information of A0 or A1 is not obtained, this module chooses this type of event if the target entity is contained in the sentence. For example, for the target entity *"Steve Jobs"* and the nominal event *"keynote"*, this event should be chosen due to the sentence in which appears: *"**Steve Jobs** gave his annual opening **keynote** on Monday"*.

Otherwise, the event is discarded.

3.1.3. Temporal Clustering

Considering the premise that two events referring to the same event happen at the same time, and using the temporal annotation of the input texts (TimeML annotation schema[14]), the temporal clustering algorithm performs two steps:

- *Within-document temporal clustering*: For each document, the temporal information of each event is extracted. Each event is anchored to a time anchor[15] when a temporal SIMULTANEOUS/ BEGIN/ INCLUDES link exists between this event and a temporal expression. After this, two events are grouped together if they are temporally compatible. This means that: a) two events are anchored to the same time anchor, or b) two events have a temporal SIMULTANEOUS link between them.

  Example 3 shows two events temporally compatible and grouped together.

  (3) a. The `<EVENT eid="e1"> meeting </EVENT>` was
  `<TIMEX3 tid="t1" value="2014-03-22"> yesterday </TIMEX3>.`

[14] http://www.timeml.org/

[15] A time anchor is always a DATE (as defined in TimeML standard annotation) and its format follows the ISO-8601 standard: YYYY-MM-DD. The finest granularity admitted in the task for a time anchor is DAY. Other granularities admitted are MONTH (references as YYYY-MM) and YEAR (references as YYYY. A time anchor takes as value the point in time when the event occurred (in case of punctual events) or began (in case of durative events). Event ordering is based on temporal relations between events; more specifically on the before/after and includes/simultaneous relations as defined by ISO-TimeML. The system places the dates in the timeline from lowest to finest granularity.

b. At the same time, the teacher `<EVENT eid="e2"> presents </EVENT>` the ideas.
`<TLINK eventInstanceID="e1" relatedToTime="t1"`
`relType="IS_INCLUDED"/>`
`<TLINK eventInstanceID="e2" relatedToEventInstance="e1"`
`relType="SIMULTANEOUS"/>`

Two non-temporally compatible events are shown in Example 4.

(4) a. The `<EVENT eid="e1"> meeting </EVENT>` was `<TIMEX3 tid="t1" value="2014-03-22T17:00"> yesterday at 17:00 </TIMEX3>`.

b. After that, the teacher `<EVENT eid="e2"> presents </EVENT>` the ideas.
`<TLINK eventInstanceID="e1" relatedToTime="t1"`
`relType="IS_INCLUDED"/>`
`<TLINK eventInstanceID="e2" relatedToEventInstance="e1"`
`relType="AFTER"/>`

- *Cross-document temporal clustering*: Considering that in the previous step all the events of each document were assigned to a time anchor, in this step, this information is merged in a single timeline, in which all the events of the different documents are grouped together if they are happening at the same time.

(5) a. Document 1: The `<EVENT eid="e1"> meeting </EVENT>` was `<TIMEX3 tid="t1" value= "2014-03-22"> yesterday </TIMEX3>`.
`<TLINK eventInstanceID="e1" relatedToTime="t1"`
`relType="IS_INCLUDED" />`

b. Document 2: The students `<EVENT eid="e5"> met </EVENT>` on `<TIMEX3 tid="t3" value="2014-03-22">Tuesday</TIMEX3>`.
`<TLINK eventInstanceID="e5" relatedToTime="t3"`
`relType="IS_INCLUDED" />`

According to Example 5 and after performing the within-document temporal clustering, doc1-e1 is anchored to the date “2014-03-22”, and doc2-e5 is anchored to the same date. Therefore, in the cross-document temporal clustering step these two events will be considered part of the same group.

Finally, the temporal groups are chronologically ordered. For each line, there is first a cardinal number indicating the position of an event in the timeline, then the value of the anchor time attribute, and finally the list of events anchored to this time attribute. Each event is represented as follows: language (en/es), document identifier, sentence number and textual extent of the event. For example, the event en-18315-7-leave is located in sentence 7 of document 18315 and it is in English. In this first clustering, if two events have the same value for the anchor time attribute, they are placed in the same group. In the next step, explained in the following section, these temporal groups will be divided again according to their semantics.

3.1.4. Semantic Clustering

Two or more event mentions in the same time slot could refer to the same real event. To detect these coreferential events, we have applied a clustering process based on two kinds of semantic information: i) the event type; and, ii) distributional semantic similarity between event mentions.

During the event extraction process, each event mention has been classified according to its type of event following TimeML standard [49]: occurrence, perception, reporting, aspectual, state, intentional state and intentional action. All the event mentions with the same time slot have been regrouped after also considering the type of event to which they have been assigned.

Next, our approach clusters coreferential events (identifies all the events that share the same time slot and the same type of event) according to the compositional-distributional semantic similarity between them. The semantics of the event structure is represented as a compositional-distributional vector. Rather than creating a complex feature matrix to represent the semantics of the argument, as described in [50], we propose a compact distributional semantic model. In this way, we consider the context of the events as the main component that contributes to establishing the semantic compatibility and, therefore, the event coreference. This relies on the fact that distributional semantics are based on the contextual meaning of words [51, 52]. Beyond trying to represent the meaning of words through lexicons or ontologies, distributional semantics represent how words are used in real context through vector spaces [53, 54]. These vectors are called contextual vectors. Specifically, for each word of the event structure we have used the English Word2Vec word embedding trained on the Google News corpus.

In our approach each event structure is formed, on the one hand by the event head and, on the other hand by the nouns, verbs and adjectives of the main arguments. All this information is extracted by applying Freeling [55] as Part of Speech tagger and Semantic Role Labeling system. Following the additive model [56], these word vectors are added in a single compositional vector that represents the distributional meaning of the whole event structure.

An event structure (ES) with two arguments is formally represented as a tuple of three elements: two arguments ($A0$ and $A1$) and one event head (H):

$$ES = < A0, A1, H > \tag{2}$$

Each argument is a compositional vector $\overrightarrow{V}(A)$ formed by the sum of the contextual vector $\overrightarrow{V}(w_n)$ of each word of the argument:

$$\overrightarrow{V}(A) = \sum^{n} \overrightarrow{V}(w_n) \tag{3}$$

where w_n represents each word of an argument and $\overrightarrow{V}(w_n)$ the contextual vector of each one of these words.

The event head H is the contextual vector of a single word. Finally, the compositional vector of the whole event structure $\overrightarrow{V}(ES)$ is:

$$\overrightarrow{V}(ES) = \overrightarrow{V}(A0) + \overrightarrow{V}(A1) + \overrightarrow{V}(H) \tag{4}$$

where + means sum of vectors.

The similarity among all vectors two-to-two is represented by a square matrix. The final cluster is obtained applying a standard hierarchical cluster to this matrix. Specifically, we have applied an agglomerative clustering based on the average linkage criteria that uses the arithmetic mean of the distances between clusters to construct the dendrogram. We consider all event mentions grouped together at level one of this hierarchical cluster, that is, the second-most coarse-grained level under the root of the dendrogram.

3.1.5. Event cluster enrichment

The timeline consists of structured information in which all the event mentions related to the same event are grouped together according to the exact date when the event occurs. However, this information is not useful if the user that needs the information only has the event core (verb or nominalization). The user will also need the arguments involved in the event to obtain the accurate information about the event. Therefore, in this step, all the arguments (semantic roles extracted in the previous step with Freeling) of the events in each cluster are added to the timeline, enriching the information provided for each event. In the Example 6, an enriched cluster of the event mentions related to the same event is presented.

(6) 0 2008 en-82548-4-built:$_{(A1,The\ plane),(A2,with\ four\ Rolls-Royce_{T}rent\ 900\ engines)}$

(EN: In 2008, they built the plane with four Rolls-Royce_Trent 900 engines)

en-82548-2-made:$_{(A1,The\ first\ A380\ superjumbo),(A0,by\ Airbus)}$

(EN: In 2008, Airbus made the first A380 superjumbo)

In the example, for each event mention, all the arguments found in the input document are added to the event mention with their corresponding semantic role (A0, A1,...). Therefore, not only the event mention is used but also the argument information.

3.2. Abstractive summarization

As previously mentioned, the aim of this module is to produce a narrative abstractive summary with information given in an enriched timeline. This summary is generated employing NLG techniques. In particular, we employ a hybrid surface realization approach, based on over-generation and ranking techniques. In these types of techniques, several possible outputs are generated and then ranked in order to select the best one, based on probability models. For each of the enriched cluster of events from the enriched timeline, the next steps are as follows:

- Argument selection: the arguments from the enriched timeline are selected in the case that there is more than one argument for the same semantic role. This selection is performed based on the probability of the phrases contained in the arguments, which is calculated using a language model.
- Obtaining verb frames: information about the frames corresponding to the verbs of each event is obtained to generate a sentence without the need to resort to grammar specifications.
- Sentence Generation: for each of the frames obtained a sentence is generated, based on the frame structure.

- Sentence Ranking: a ranking is performed for selecting only one sentence representing a specific event (cluster of event mentions) in the timeline.

Before beginning the generation process, a language model is trained over each of the input documents. This language model will be employed in some of the steps of this module, and in particular, Factored Language Models (FLM) are used to train it. FLM are an extension of the conventional language models, proposed in [57], where a word is viewed as a vector of k factors such that $w_t \equiv \{f_t^1, f_t^2, \ldots, f_t^K\}$. The factors within this kind of model can be anything, ranging from more basic elements, such as words or lemmas to any other lexical, syntactic or semantic features needed for the task to be addressed. The main objective of this type of model is to create a conditional probability model over the selected factors: $P(f|f_1, \ldots, f_N)$, being the prediction of the factor f based on its N parents $\{f_1, \ldots, f_N\}$. For the purpose of this research, information about words, lemmas, Part-of-Speech (POS) tags and synsets[16] are used as the factors for training the FLMs. These factors were selected due to the type of information they provide. In this regard, syntactic and semantic information along with information about the words themselves are needed in order to create a flexible abstractive summary in relation to its vocabulary. To deal with these types of statistical models, the SRILM [58] is used. This software is a toolkit for building and applying statistical language model, which includes an implementation of FLM.

3.2.1. Argument selection

Taking as input the enriched timeline, for each of the events contained in it, their arguments are checked to avoid duplicate semantic roles in the same event.

In the case that two or more arguments for the same semantic role appear within the event, the probability of the phrases contained in the arguments is calculated employing the FLM previously trained. This probability is calculated employing only the words in the arguments either using the probability given by the FLM when the phrase has 3 or less words, or otherwise, using the chain rule (see Equation 5). In the chain rule, the probability of a phrase

[16] Set of cognitive synonyms related to a concept used in WordNet.

or a sentence is calculated as the product of the probability of all its words.

$$P(w_1, w_2...w_n) = \prod_{i=1}^{n} P(w_i|w_1, w_2...w_{i-1}) \tag{5}$$

When the probability of the different arguments for the same semantic role is calculated, the argument with the highest probability is selected. In Example 7 an event with several arguments for the same semantic role is shown. In this example, the first argument for A1 (i.e. *Boeing*) will be selected since its probability is higher than the one of the second argument for A1 (i.e. *Civilian Deputy Undersecretary Darleen Druyun*).

(7) 0 2005 en-1173-35-hired:$_{(A1,Boeing),(A1,CivilianDeputyUndersecretaryDarleenDruyun)}$
Probability of "Boeing": 0.20
Probability of "Civilian Deputy Undersecretary Darleen Druyun": 0.15

3.2.2. Verb frame extraction

After the different elements of the enriched timeline (i.e. their arguments) are selected, the lexical resources VerbNet [59] and WordNet [60] are used to obtain syntactic frames, from their event cores, which will be used during the summary generation. VerbNet is one the largest verbs lexicons for English including semantic and syntactic information about verbs. WordNet is a lexical database composed by sets of synonym elements. Using both resources, a set of frames containing the following information is extracted: i) the frames from VerbNet comprise syntactic as well as semantic information about each of the verbs of the lexicon; ii) WordNet provides a set of generic frames for all the verbs. For every event, a set of frames from both, VerbNet and WordNet are compiled. These frames are then analyzed to find out which elements of the sentences need to be generated in the next step —the components of the sentence, such as the subject or the object—. This avoids having to define a grammar specification with the associated high cost.

When extracting the frames from VerbNet and WordNet, the "V" in the frames from Verbnet represents the verb. WordNet, in this regard, is used to extract the generic frames from a verb, which are consequently used to produce a sentence for each of them.

Example 8 shows the frames which would be obtained from the event cores of the Example 6 (i.e. *built* and *made*). Since the verbs *build* and *make*, for the sense of constructing something combining materials and parts, belong to the same VerbNet class and have the same synset in WordNet, the extracted frames are the same for both.

(8) **VerbNet frames**
Agent V
Agent V Material
WordNet frames
Somebody - - - -s something

3.2.3. Sentence generation

For each of the frames obtained in the previous step, a sentence is generated. If the specific event from which the verb frame was extracted has arguments, the sentence is generated using these arguments along with the information from the verb frame. The components of the frame may indicate the need for some particular type of semantic role, such as an agent (i.e. A0, A1) or an instrument (i.e. A2). Therefore, the sentence will be composed using only the arguments needed and putting them in the order specified by the frame. In certain cases, where the verb permits, if there is not an A0 but an A1 argument, the A1 is treated as the Subject of the sentence, and this sentence is generated in the passive voice.

In the case that the event does not have any arguments, a sentence is generated following the structure given by the verb frame. For instance, if the frame indicates the need for a Subject, it is generated based on the FLMs trained, choosing the words with the highest probability appearing with the corresponding verb of the event. The Object of the sentence is generated using the same process, if needed.

In Example 9 the generated sentences for the frames shown in Example 8 can be seen. It is possible that, for the same verb, the frames obtained from VerbNet and WordNet contain similar information to decide which arguments of the event to select. In these cases, it is likely that the sentences generated by both frames are the same, since they use the same arguments to generate it.

(9) **build**
The plane was built.
The plane was built with four Rolls-Royce Trent 900 engines.
The plane was built with four Rolls-Royce Trent 900 engines.
make

by Airbus made.
by Airbus made the first A380 superjumbo, made by Airbus.
by Airbus made the first A380 superjumbo, made by Airbus.

3.2.4. Sentence ranking

Once a set of possible sentences containing the information of a specific event is generated, a ranking is performed in order to select the sentence which will form part of the chronological abstract summary. For selecting the final sentences, the following process is applied: sentences are ranked based on their probability which is computed by the chain rule (see Section 3.2.1).

The calculation of the probability of a word may differ depending of the language model employed. Since, in this work, FLMs are used, the probability of a word is calculated as the linear combination of FLMs as suggested in [61] where a weight λ_i, is assigned to each of them (see Equation 6), being their total sum 1. In this Equation, f refers to a lemma, p refers to a POS tag, and λ_i are set $\lambda_1 = 0.25$, $\lambda_2 = 0.25$ and $\lambda_3 = 0.5$. These values were empirically determined by testing different values and comparing the results obtained.

$$P(w_i) = \lambda_1 P(f_i|f_{i-2}, f_{i-1}) + \lambda_2 P(f_i|p_{i-2}, p_{i-1}) + \lambda_3 P(p_i|f_{i-2}, f_{i-1}) \qquad (6)$$

The final selected sentence will be the one with the highest probability. This sentence along with the date on which the event took place will be considered as the sentence representing the information of the event.

Example 10 shows the final sentence selected from the ones in Example 9. The probabilities provided for each sentence are computed employing the chain rule explained above (Equation (6)).

(10) Probability of “The plane was built.” : 0.16
Probability of “The plane was built with four Rolls-Royce Trent 900 engines.”: 0.25
Probability of “The plane was built with four Rolls-Royce Trent 900 engines.”: 0.25
Probability of “by Airbus made.”: 0.12
Probability of “by Airbus made The first A380 superjumbo, made by Airbus.”: 0.08
Probability of “by Airbus made The first A380 superjumbo, made by Airbus.”: 0.08
Final Selected Sentence: The plane was built with four Rolls-Royce Trent 900 engines.

Then, this sentence will be included in the final narrative abstractive summary together with the remaining sentences generated by repeating this process for each line in the enriched timeline.

4. Experimental Setup and Evaluation

NATSUM is focused on the transformation from a simple timeline to a coherent narrative abstractive summary. For the evaluation of our system, the test dataset provided for Task 4 at SemEval 2015 is used.[17] This dataset is composed of Wikinews articles about different topics: Airbus and Boeing; General Motors, Chrysler and Ford; and the Stock Market. This evaluation corpora consists of 90 documents (around 30,000 tokens and 915 events) and they are very similar in terms of size. Each narrative abstractive summary generated from the enriched timeline is entity-focused. This means that a set of target entities is also provided within the corpus, and each timeline is only composed of events related to this target entity. There is a total of 35 target entities in this dataset.

The following subsections provide information about the main experiments carried out with the SemEval 2015 Task 4 dataset (Section 4.1), and the evaluation methodology proposed (Section 4.2).

4.1. Main Experiments

Regarding the experiments conducted, for each target entity in the SemEval 2015 Task 4 dataset, a narrative abstractive summary was generated considering two configurations: (i) gold-standard experiment and (ii) overall system experiment. In total, 70 narrative summaries were generated (35 summaries for each experiment). For the gold-standard experiment, gold-standard timelines provided in SemEval 2015 Task 4 are used. Using these gold-standard timelines it is possible to measure the abstractive summarization module, avoiding the errors derived from the enriched timeline generation task. For the overall experiment, unannotated data is used to evaluate the system in a real scenario in which our narrative abstractive summaries could be applied. In this manner, the raw data of the Semeval corpus was used as input, and then, the Enriched Timeline Extraction module provided an intermediate scheme. The scheme contains the events and temporal information to be used by the Abstractive Summarization module to generate

[17]http://alt.qcri.org/semeval2015/task4/index.php?id=data

the sentences that will compose the final narrative summary. Furthermore, the Timeline Extraction module was evaluated in isolation obtaining the following results for English: F1-measure 27.63%, Precision 25.28%, Recall 30.47%. These results surpass the evaluation presented in [45], but evaluating the Enriched Timeline Extraction module is beyond the scope of this work. In addition, several state-of-the-art extractive summarization systems were also used for the experiments for comparison purposes. In particular, we selected the following systems: COMPENDIUM [62], GRAFENO [63] and Open Text Summarizer (OTS) [64], since they provide either a visual interface or the program to generate the summaries. In order to generate multi-document and entity-focused extractive summaries that contain the relevant information about a given entity, the input documents were preprocessed following a two-step strategy. Firstly, all the documents belonging to the same corpus were merged into a single macro-document; and secondly, noisy sentences were removed from the input macro-document, i.e., the sentences not talking about the focused entity or referring to them. By this means, the job of summarization systems was only focused on determining the relevant information to generate the final extractive summary, so the techniques they implemented remained the same. In the end, 35 summaries were produced by each system.

Finally, two baselines for narrative abstractive summarization were also proposed (*FirstEvent* and *LongestEvent*). These baselines generate the narrative summary using either only the first event (*FirstEvent*), or the event with the highest number of arguments (*LongestEvent*) of each cluster provided by the gold-standard timelines—for experiment (i)—, or by the enriched timeline— for experiment (ii)—.

4.2. Evaluation Methodology

To assess the appropriateness of the resulting summary in terms of its content and fluency, two types of quantitative evaluation were performed, together with an additional human linguistic evaluation.

The first quantitative evaluation involved the analysis of extractive summaries generated by state-of-the-art summarization systems. The goal of this evaluation was to determine to what extent extractive summarization systems were able to capture the relevant events and temporal information contained in the input documents, and whether these systems were appropriate for conducting narrative summarization or not. For this, we computed the number of events and temporal information, comparing them to the gold-standard annotations of the corpus employed. In order to avoid the errors that may be obtained by just computing whether an event is present or not in the summary we also took into account the location of the event, i.e., the sentence in which it appears. For instance, the summary may contain a verb but this does not necessarily refer to the same event of the gold-standard, underscoring the importance of identifying the context in which the event occurred so as to verify the accuracy of the generated summary.

The second type of quantitative evaluation is based on the hypothesis that our abstractive summarization proposal enhances the quality of the narrative summaries, relying on NLG techniques and using temporal information. For this purpose, ROUGE tool [65] was used. ROUGE evaluates how informative an automatic summary is by comparing its content to one or more reference summaries. Such comparison is made in terms of n-gram co-occurrence (e.g., unigrams, bigrams, or word sequences). Moreover, ROUGE implements different metrics, such as unigram similarity (ROUGE-1); bigram similarity (ROUGE-2); longest common subsequence (ROUGE-L) and bigram similarity skipping unigrams (ROUGE-SU4). For each of these metrics, it provides the commonly used HLT measures (precision, recall and F1-measure):

$$Precision = \frac{\#CorrectPhrasesExtracted}{\#TotalPhrasesExtracted}, \tag{7}$$

$$Recall = \frac{\#CorrectPhrasesExtracted}{\#CorrectPhrasesTest}, \tag{8}$$

$$F1 - measure = \frac{2 * Precision * Recall}{Precison + Recall}, \tag{9}$$

where *#CorrectPhrasesExtracted* is the number of correct sentences that the evaluated system extracts, *#TotalPhrasesExtracted* the total number of sentences that the evaluated system extracts and *#CorrectPhrasesTest* the total number of sentences included in the reference summaries.

ROUGE requires reference summaries and the creation of them is a time-consuming and costly task. Therefore, a semi-automatic process was implemented in order to generate a reference summary directly created from the gold-standard timelines that were available within the corpus used for the experiments. This process is further described in Section 4.2.1.

After having created the set of reference summaries, we directly compared the content of the generated summaries to the reference ones. For this evaluation, apart from our proposed narrative abstractive summarization approach (NATSUM) , we also considered the extractive systems previously analyzed (COMPENDIUM, GRAFENO and OTS), as well as the two proposed baselines (FirstEvent and LongestEvent). This enabled a comparison of this paper's proposal with other approaches, as well as verifying whether extractive summarization systems present limitations when it comes to performing this task.

Using ROUGE for conducting this evaluation is appropriate as the events are represented with words (generally verbs). Therefore, if the automatic summary correctly captures the relevant events together with the right arguments, the result for the ROUGE metrics will increase because the generated summary and the reference summary (gold-standard) are similar. In this context, the summaries contain the key information of the documents. However, using ROUGE exclusively for the evaluation is limited, since it is not useful for determining the linguistic quality of the generated summaries and is incapable of deciding the degree of grammatical correctness and meaningfulness of the summaries. In this manner, a human evaluation was also carried out involving several assessors that evaluated the linguistic quality of the generated summaries. Hence, quantitative as well as qualitative results were obtained (reported and explained in Section 5). The linguistic quality of the generated abstractive summaries was assessed taking the readability and linguistic criteria of the well-known summarization tracks for DUC[18] and TAC[19] conferences as a benchmark. Specifically, we evaluated the readability/fluency of the summaries, including different criteria, such as the summary's grammaticality, non-redundancy, referential clarity, focus, as well as structure and coherence. Moreover, the summary's overall responsiveness was also evaluated to determine the extent to which the amount of information in the summary actually helped satisfy the information requirement.

For this, 12 humans with an advanced level of English participated in this evaluation. The task consisted of completing a questionnaire[20] that tackled the previously mentioned linguistic issues. Finally, also as part of the manual evaluation, a human relevance judgement evaluation was carried out. In this manner, we could check from a human perspective, which system

[18]https://www-nlpir.nist.gov/projects/duc/index.html
[19]https://tac.nist.gov/
[20]https://goo.gl/buC68B

generated the summaries that were most preferred by users. To conduct this task, assessors had to assign a preference ranking for a set of summaries, indicating their most preferred, second most preferred and least preferred summary. A second questionnaire was designed for this purpose[21].

4.2.1. Generation of reference summaries

In this section, we explain the process for creating the reference summaries that will be used in the quantitative evaluation. To create reference summaries that allow us to evaluate the proposal, a set of patterns are applied over the gold enriched timelines.

The following steps are performed in order to generate each sentence that will compose the reference summary:

- *Verb selection*: Since the cluster contains different event mentions for the same event, in the reference summary the first verb in the cluster is used as representative of all the events in the cluster.
- *Arguments selection*: In order to create the sentence, only one of each type of argument is necessary. In case there is more than one, the longest one is chosen, since it is the most complete one, and it would contain more information about the argument, thus leading to a more informative sentence.
- *Sentence generation*: For each cluster, a sentence following this pattern is generated:

  (11) **Pattern**: *Time* A0 *event* A1 A2 A3 A4

  Only the arguments available are used. A2, A3 and A4 are optional, but in case there is no A0, or A1, the target entity is used.

  In case of nominalizations, since they are not verbs, it is not possible to obtain any semantic role. For these cases, we create a sentence using the pattern:

  (12) **Pattern**: *Time TargetEntity* had a *NominalizationEvent*
  Example: On February *(Time)* Airbus *(TargetEntity)* had a crush *(Nominalization)*

[21] https://goo.gl/Mrj8yY

5. Results and Discussion

In this section, we show the results obtained through the different evaluations described in the previous section, as well as the analysis of these results.

5.1. Limitations of Extractive Summarization

Table 1 shows the results obtained after analyzing both the number of relevant events and the presence of temporal information that were contained in the extractive summaries generated by COMPENDIUM, GRAFENO and OTS. As observed, although the extractive summarization systems were adapted to be multi-document and entity-focused, they are only able to capture a small percentage of the relevant events and temporal information that should be included in the narrative summary. Concerning the number of events reflected in the summary, the highest result was obtained by the GRAFENO system (38.49%), but this result still represents less than half of the relevant events identified in the gold-standard. As for the temporal information, we noted that GRAFENO is the extractive system that obtains the poorest results, reflecting 7% of the temporal information, which may render difficult the comprehension of the summary with respect to the dates of the different events. COMPENDIUM and OTS, the other systems used, both exhibit similar performance.

Given that several relevant events were not captured and temporal information was omitted— hence, these items were not extracted as part of the output summary— we can conclude that traditional extractive summarization systems are not effective in terms of generating narrative summaries.

System	**Events**	**Temporal information**
COMPENDIUM	26.86%	18.90%
GRAFENO	38.49%	7.10%
OTS	22.04%	18.04%

Table 1: Average percentage of events and temporal information reflected in extractive summaries.

5.2. Summarization Results

This section describes the automatic and manual evaluation for NATSUM within the two experiments conducted: i) gold-standard experiment, and ii) overall system experiment. Section 5.2.1 specifically reports the results obtained after automatically evaluating the content of summaries using ROUGE tool, whereas Section 5.2.2 provides the results for the manually conducted linguistic and readability evaluation. For both subsections, we also compare NATSUM with respect to other summarization systems and baselines.

5.2.1. Automatic Evaluation

The results shown in this section refer to the content assessment of the narrative summaries generated by NATSUM compared to reference summaries. As previously stated in Section 4, ROUGE was selected as the tool for automatically evaluating our summaries, since it is a widespread summarization evaluation tool that has been shown to correlate well with human evaluations [66]. The most recent version of ROUGE (ROUGE-1.5.5) was used.

Table 2 and Table 3 report the average ROUGE recall (R), precision (P) and F1-measure (F) for the following metrics: ROUGE-1 and ROUGE-2—compute the number of overlapping unigrams and bigrams, respectively—; ROUGE-L—calculates the longest common subsequence between an automatic and a reference summary—; and, ROUGE-SU4—measures the overlap of skip-bigrams an automatic summary contains with respect to a model one, with a maximum distance of four words between them—. The higher the recall, precision and F1-measure values, the better.

The two tables differ in the input given for the Abstractive Summarization module corresponding to the experimental scenarios described in Section 4.1: i) the gold-standard, and ii) the overall experiment, respectively. Whereas in Table 2, the input for this module is derived from the gold-standard timelines available in the corpus, Table 3 reports the results of the system in a real scenario, thus allowing us to also analyze how the overall system performs.

Furthermore, the "FirstEvent" refers to the narrative summary approach generated, only taking into account the first event provided by the enriched timeline, which is considered as a baseline. The "LongestEvent" refers to an additional narrative summarization approach that takes into account, for each line of the given timeline, the event with the higher number of arguments, to generate a sentence from it. We also computed the performance of the extractive summarization approaches previously analyzed (COMPENDIUM, GRAFENO, OTS).

	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4		
	R	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
COMPENDIUM	0.317	0.370	0.312	0.114	0.154	0.121	0.296	0.348	0.293	0.142	0.180	0.145
GRAFENO	0.285	0.415	0.295	0.102	0.199	0.118	0.261	0.384	0.272	0.127	0.140	0.139
OTS	0.305	0.362	0.303	0.106	0.148	0.114	0.280	0.335	0.280	0.133	0.173	0.138
FirstEvent	0.323	0.583	0.402	0.141	0.270	0.179	0.316	0.570	0.392	0.140	0.264	0.176
LongestEvent	0.351	0.688	0.445	0.166	0.335	0.215	0.340	0.665	0.431	0.165	0.339	0.214
NATSUM	**0.576**	**0.735**	**0.637**	**0.420**	**0.544**	**0.467**	**0.559**	**0.714**	**0.619**	**0.400**	**0.518**	**0.445**

Table 2: Average values for recall, precision and F1-measure for the gold-standard annotations ((i) gold-standard experiment). Comparison between different summarization and baseline approaches.

	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4		
	R	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
COMPENDIUM	0.317	0.370	0.312	0.114	0.154	0.121	0.296	0.348	0.293	0.142	0.180	0.145
GRAFENO	0.285	0.415	0.295	0.102	0.199	0.118	0.261	0.384	0.272	0.127	0.140	0.139
OTS	0.305	0.362	0.303	0.106	0.148	0.114	0.280	0.335	0.280	0.133	0.173	0.138
FirstEvent	0.258	0.463	0.302	0.083	0.164	0.101	0.250	0.444	0.293	0.100	0.194	0.119
LongestEvent	0.251	0.524	0.312	0.088	0.196	0.114	0.245	0.510	0.305	0.099	0.225	0.125
NATSUM	**0.433**	**0.595**	**0.470**	**0.263**	**0.363**	**0.284**	**0.422**	**0.579**	**0.457**	**0.260**	**0.360**	**0.282**

Table 3: Average values for recall, precision and F1-measure when using raw data without any type of annotation as input ((ii) overall system experiment). Comparison between different summarization and baseline approaches in a real scenario.

For each table, rows 3-5 refer to the extractive summarization approaches, whereas rows 6-8 refer to abstractive summarization. The results indicate that regardless of the input type used for the Abstractive Summarization module (either the gold-standard timelines for event identification available in the corpus, or the ones produced by the Enriched Timeline Extraction module), our system outperforms the remaining ones. This means that integrating the module for identifying events, as well as extracting temporal information enhances narrative summarization. When the complete system is evaluated, the results for the last two rows in Table 3 are lower than the corresponding ones in Table 2. This is explained by the errors that the Enchiched Timeline Extraction module may introduce in the overall system. However, despite this issue, in both evaluations, NATSUM obtains better results than the others.

Table 4 and Table 5 provide the percentage of improvement obtained by NATSUM compared to the remaining summarization systems and baselines, taking only into account the F1-measure values.

	COMPENDIUM				GRAFENO				OTS				FirstEvent				LongestEvent			
	R1	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**
NATSUM	105	286	111	207	116	295	128	220	110	309	121	223	59	160	58	153	43	117	43	108

Table 4: Percentage of improvement for the F1-measure metric when comparing NATSUM with respect to the extractive summarization approaches and abstractive baselines for the gold-standard annotations ((i) gold-standard experiment). R1, R2, RL, and RSU4 refer to ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-SU4, respectively.

	COMPENDIUM				GRAFENO				OTS				FirstEvent				LongestEvent			
	R1	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**
NATSUM	51	135	56	95	59	140	68	103	55	149	65	105	56	182	56	137	51	153	50	125

Table 5: Percentage of improvement for the F1-measure metric when comparing NATSUM with respect to the extractive summarization approaches and abstractive baselines when using raw data without any type of annotation as input ((ii) overall system experiment). R1, R2, RL, and RSU4 refer to ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-SU4, respectively.

The results indicate that NATSUM performs better than other summarization approaches. This improvement is even greater when compared to the extractive summarization approaches. Moreover, despite the LongestEvent baseline being more competitive than the FirstEvent baseline, NATSUM is still capable of delivering a better performance. On the one hand, when considering the gold-standard timelines (i.e., only the Abstractive Summarization module without using the Enriched Timeline Extraction module), NATSUM's performance increases for the F1-measure by 59% for ROUGE-1; 160% for ROUGE-2; 58% for ROUGE-L; and 153% for ROUGE-SU4 compared to the FirstVerb baseline; and by 43% for ROUGE-1; 117% for ROUGE-2; 43% for ROUGE-L; and 108% for ROUGE-SU4 compared to the LongestEvent baseline. On the other hand, when considering the raw data without any kind of annotation as input—i.e. our complete approach, integrating both modules explained in Section 3—, NATSUM's performance is also increased compared to the baselines as can be seen in Tables 3 and 5.

NATSUM also performs better than the multi-document entity-focused extractive summarization tested. The extractive summarization system with the best F1-measure results for all ROUGE metrics —COMPENDIUM— is improved by 51% for ROUGE-1, when our narrative abstractive approach is compared to the best extractive summarization system in the real scenario— i.e., with raw text as input data for the approach without any type of annotation on events—. When gold-standard timelines are considered, this improvement increases by 105% for ROUGE-1.

Additionally, the use of NLG techniques does not decrease the performance of the resulting summaries, as demonstrated by the results of Table 2, when the input for the Abstractive Summarization module comes from gold standard event and temporal annotations, thus indicating that NLG can benefit abstractive summarization. This reconfirms our initial claim that extractive summarization is not sufficient for generating effective narrative summaries.

Finally, the main conclusion of this quantitative evaluation using ROUGE is that NATSUM's approach of integrating the Enriched Timeline Extraction module for identifying co-referent events and temporal information in different related documents, together with an Abstractive Summarization module using NLG techniques is highly effective for producing narrative summaries.

In Example 13, a fragment of a generated narrative abstractive summary about "Boeing" using our NATSUM system is shown.

(13) 2006-01: The first of the new airliner delivered to Pakistan International Airlines.
2007-06-10: The aircraft have a pre-modification catalogue value of US $ 3.5 billion.
2007-07-07: Announced 35 new orders from German airline Air Berlin and ALAFCO Aviation Lease & Finance of Kuwait.
2007-07-08: Boeing received a congratulatory letter from Airbus.
2007-07-08: The plane promises as it is the first model to be built out of plastic and carbon composites, more lightweight than conventional materials.

5.2.2. Readability Evaluation

This section reports the results obtained for the manual readability evaluation. As previously explained in Section 4, a linguistic evaluation with human assessors was also conducted to determine whether the abstractive summaries were appropriate from a readability perspective.

For this evaluation, we only compared the abstractive summaries, NATSUM and the two baselines — FirstEvent and LongestEvent— since they used NLG techniques to create the summaries. Therefore, to verify the linguistic quality of the generated content was more critical in this case, whereas extractive summaries just copy and paste the same content available from the original documents.

Table 6 and Table 7 report the average results obtained for i) the gold-standard, and ii) the overall experiment, respectively.

	Readability/Fluency						Overall
	Grammaticality	Non-redundancy	Referential clarity	Focus	Structure and Coherence	Average	Responsiveness
FistEvent	2.47	2.70	2.73	2.42	1.97	2.46	2.16
LongestEvent	2.08	2.77	2.80	2.30	1.85	2.36	2.03
NATSUM	2.78	3.18	3.36	3.25	2.83	3.08	2.89

Table 6: Average values for readability/fluency (including the average values for summary's grammaticality, non-redundancy, referential clarity, focus and structure and coherence) and for the summary's overall responsiveness for the (i) gold-standard experiment.

	Readability/Fluency						Overall
	Grammaticality	Non-redundancy	Referential clarity	Focus	Structure and Coherence	Average	Responsiveness
FistEvent	2.52	2.81	2.84	3.00	2.33	2.70	2.74
LongestEvent	2.45	2.76	3.05	2.90	2.21	2.67	2.66
NATSUM	2.69	3.41	3.53	3.79	3.07	3.30	3.60

Table 7: Average values for readability/fluency (including the average values for summary's grammaticality, non-redundancy, referential clarity, focus and structure and coherence) and for the summary's overall responsiveness for the (ii) the overall system experiment.

As can be seen in the tables, in both experiments NATSUM obtains better results than the ones obtained by the two baselines. These results indicate that NATSUM improves the linguistic quality of the generated summaries in comparison to the baselines, thus corroborating the results achieved in the automatic evaluation. In terms of readability/fluency results, the summaries generated by NATSUM have a higher structure and coherence than the baselines summaries. In addition to this, they present less redundancy and more referential clarity as well as more grammaticality than the ones from the baselines, maintaining a better focused summary. Moreover, in terms of overall responsiveness, NATSUM summaries have scored higher for both experiments.

Furthermore, as mentioned, a human relevance judgement evaluation was carried out. In this case, the assessors preferred the summaries generated by NATSUM for both experiments –79.45% and 79.66% for the gold-standard and overall experiments, respectively–.

6. Assessing NATSUM in the context of Timeline Summarization

To the best of our knowledge, there is no specific dataset with reference summaries that could be appropriate for the specific features of NATSUM (i.e., narrative chronological abstractive summarization). However, having obtaining good results in the evaluation conducted in Section 4.2, it would be also important to validate these results and findings by benchmarking NATSUM against additional existing datasets developed for a similar task (i.e., timeline summarization). Besides the comparison with the extractive systems already used throughout this research work (i.e., COMPENDIUM [62], GRAFENO [63] and Open Text Summarizer (OTS) [64]), this would allow us to compare NATSUM with more task-oriented and focused state-of-the-art systems.

Summaries generated for the task of timeline summarization mainly differ from those generated by NATSUM in that the latter aims to generate narrative summaries and not timelines. In the case of NATUSM, timelines constitute the means to generate the final narrative structure. In this sense, the input of the abstractive summarization module is not a set of documents, but a target oriented timeline. In contrast, in the case of timeline summarization, the final aim is to generate a timeline that serves as the summary of one or more input documents.

Regardless of these differences, and considering that the final timelines in timeline summarization contain short summaries temporally ordered by the document creation time, NATSUM is evaluated using an specific available dataset for the task of timeline summarization. The dataset finally chosen for the evaluation and comparison is Timeline17 dataset, which is the one used in [43] and [44]. The reasons for using this dataset were twofold. On the one hand, it was selected because it is available online[22] and, on the other hand, a comparison with other timeline summarization systems is presented as well. Therefore, using the same dataset, the ultimate goal of this evaluation is to compare NATSUM with all the timeline summarization systems presented in [43] and [44], as well as compared it with the extractive multi-document summarization systems presented throughout this research work (COMPENDIUM, OTS and GRAFENO) to confirm and validate whether the summaries generated by NATSUM offer an added value with respect to a standard timeline extractive summary.

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

In the next subsections, we describe the dataset in more detail (Section 6.1) together with the results obtained (Section 6.2).

6.1. Timeline17 Dataset Description

This dataset is composed of news articles from different media outlets about 9 different topics: BP Oil, Michael Jackson Death, H1N1, Haiti Earthquake, Financial Crisis, Libyan War, Iraq War, Egyptian Protest, and Syrian Crisis. The dataset, created by the authors of [43] and [44], was gathered in two steps:

- Collecting human timelines (ground truth): They collected available timelines published by popular news agencies such as CNN, BBC, NBC-news, etc. that discuss the previous 9 topics. From these topics, 17 timelines were manually built. This human timelines are the gold standard (i.e., reference summaries) for the evaluation performed in the next section.

- Retrieving news articles: For each timeline, they used Google Web Search Engine[23] to retrieve news articles from the same news agency of the timeline (i.e. BBC news articles for BBC-published timeline,...) using topics as query. In the end, they obtained 4,650 news articles after removing duplicate news. All these news articles are the input to NATSUM system.

6.2. Results and Comparison with Timelime Summarization Systems

In order to apply NATSUM to the timeline summarizaton dataset described in the previous section, the system needs to use the different topics as target entities for each timeline generated (BP Oil, Michael Jackson Death, H1N1, Haiti Earthquake, Financial Crisis, Libyan War, Iraq War, Egyptian Protest and Syrian Crisis). Then, the two modules of the proposal are applied to the input documents to create the different narrative abstractive summaries. Once the summaries were generated, they were evaluated with ROUGE with respect to the reference timeline summaries available in the dataset. In order to evaluate the summaries under the same conditions, ROUGE was set to truncate the length of the generated summaries to the same length as the reference timelines had.

[23]https://www.google.com/

Table 8 reports the average F1-measure (F) results for ROUGE-1, ROUGE-2 and ROUGE-SU4 results. Rows 3-5 refer to the performance of the extractive summarization approaches previously analyzed (COMPENDIUM, GRAFENO, OTS), whereas rows 6-10 refers to the timeline summarization systems presented in [43] and [44]. Finally, the last row provides NATSUM performance[24].

	ROUGE-1	**ROUGE-2**	**ROUGE-SU4**
	F	**F**	**F**
COMPENDIUM [62]	0.340	0.085	0.133
GRAFENO [63]	0.267	0.069	0.102
OTS [64]	0.337	0.076	0.127
Chieu et al.[39]	0.202	0.037	0.041
MEAD[67]	0.208	0.049	0.039
ETS[40]	0.207	0.047	0.042
Tran Linear Regression[44]	0.218	0.050	0.046
Tran LTR[43]	0.230	0.053	0.050
NATSUM	**0.413**	**0.121**	**0.176**

Table 8: Average F1-measure values when using Timeline17 dataset as input. Comparison between different multi-document and timeline summarization approaches.

As shown in Table 8, NATSUM greatly overperforms timeline summarization systems for all ROUGE metrics, being the main reason that the summarization module is using an enriched timeline as input. The approach exploits not only the temporal information about the document creation time (as timeline summarization does) but also all the temporal links and expressions related to the events referring to the target entity across different documents. This implies a temporal information processing that goes further in terms of exploiting temporal information than merely using the document creation time. Furthermore, NATSUM approach is using the events in the timeline, and their arguments, to generate a sentence that covers all the arguments of the event. Since NATSUM is dealing with the coreference of events, for the same event, named in different ways in different documents, our final

[24]Only F1-measure for ROUGE-1, ROUGE-2, and ROUGE-SU4 is presented since this is the measure reported in referenced papers.

summary is generating a single sentence which condenses all the information related to the event in question, which results in avoiding redundancy in the resulting summary. Furthermore, the results obtained corroborate the previous evaluation of NATSUM in comparison with extractive multi-document summarization systems. Despite using a different input corpora, NATSUM performs better than COMPENDIUM, GRAFENO and OTS. It is also worth noting that extractive summaries obtain higher ROUGE results than timeline summaries. This could be explained by the fact that those systems are very competitive as far as detecting relevant information from input documents is concerned.

Finally, the results also indicate that providing a narrative abstractive summary instead of just a timeline summary is better, since besides including dates, they also provide relevant information that is generate from the information found in different sources about the same event. This validates the appropriateness of the NLG techniques used within the NATSUM system for generating abstractive summaries.

7. Conclusions

This work presents NATSUM, a narrative abstractive summarization approach that integrates structured timeline knowledge together with natural language generation techniques to enhance the creation of such type of summaries. Our integrated approach was motivated by two aspects: First, it is based on the fact that humans tend to apply chronological ordering of events in the summarizing process, which implies the need for timelines. Second, when using an abstractive summarization approach, rather than an extractive one, the relevant information (e.g., *who? what?, when?, where?,...*) can be fused together, leading to the generation of more complete sentences, and thus, more comprehensible and effective summaries. Hence, NATSUM's architecture comprises two main modules: i) Enriched Timeline Extraction module, and ii) Abstractive Summarization module. The former module uses a set of plain news documents and a target entity as input, and obtains a structured timeline document plan that is enriched with all the arguments of each event involved in the timeline for the particular target entity. Specifically, for each line of the timeline, there is a cluster with the exact date of the event and a set of event mentions together with their arguments, extracted from different documents, that refer to the same event. The latter module generates a narrative abstractive summary using the enriched timeline. For this, a hybrid surface realization approach, based on over-generation and ranking techniques is used.

The evaluation conducted and the results obtained show that extractive summaries lose between 22% (OTS) and 38%(GRAFENO) of the *events* related with the target entity; and between 7% (GRAFENO) and 19% (COMPENDIUM) of the *temporal information*. Moreover, regarding the content evaluation of the narrative abstractive summaries, the F1-measure for all ROUGE metrics improves by at least 50% in the worst case, when our narrative abstractive system (NATSUM) is compared to the extractive summarization systems, as well as to the baselines in the real scenario—i.e., with raw text as input data for the approach without any type of annotation about events—. Remarkable improvements are also obtained for the gold-standard experiment.

In addition, a manual evaluation was carried out between the summaries generated by the two baselines and NATSUM to measure the readability/fluency and overall responsiveness of the summaries. The results obtained corroborate the ones from the automatic evaluation, with the summaries from NATSUM being better than both of the baseline ones for both experiments ((i) gold-standard and (ii) overall experiments). Besides, a human relevance judgement evaluation was performed, where the NATSUM summaries were preferred in almost 80% of the cases for both experiments. Finally, in order to compare NATSUM with other systems, a timeline summarization dataset is used as input, since it is the most similar task to our proposal, concluding that NATSUM greatly improves the results obtained by state-of-the-art timeline summarization and extractive systems.

Although NATSUM has shown very good and promising results, also improving the performance of extractive summarization approaches, there are several aspects to consider for future development concerning the individual modules that are integrated into NATSUM. First, the Enriched Timeline Extraction module should be improved to better identify co-referent events and temporal relationships between events, especially when these relationships are implicit. This would narrow the gap between the results obtained when using gold-standard timelines. Second, the Abstractive Summarization module should be improved so that it would include appropriate discourse markers for connecting individual sentences to increase the coherence of the produced narrative summaries, rather than listing a set of relevant newly generated sentences. This would enhance the quality of the resulting narrative summaries generated by NATSUM.

Acknowledgements

This research work has been partially funded by the Spanish Government through projects TIN2015-65100-R, TIN2015-65136-C2-2-R, as well as by the project "Analisis de Sentimientos Aplicado a la Prevencion del Suicidio en las Redes Sociales (ASAP)" funded by Ayudas Fundacion BBVA a equipos de investigacion cientifica. Moreover, it has been also funded by Generalitat Valenciana through project "SIIA: TecnologÃas del lenguaje humano para una sociedad inclusiva, igualitaria, y accesible" with grant reference PROMETEO/2018/089

References

[1] E. Lloret, M. Palomar, Text summarisation in progress: A literature review, Artif. Intell. Rev. 37 (2012) 1–41.

[2] I. Mani, Advances in Automatic Text Summarization, MIT Press, Cambridge, MA, USA, 1999.

[3] J. Gottschall, The Storytelling Animal, Houghton Mifflin Harcourt, 2012.

[4] M. R. Hovav, E. Doron, I. Sichel, Lexical Semantics, Syntax, and Event Structure, Oxford University Press, Oxford, 2010.

[5] I. Mani, J. Pustejovsky, R. Gaizauskas, The Language of Time, Oxford University Press, Oxford, 2005.

[6] H. Ji, R. Grishman, Z. Chen, P. Gupta, Cross-document event extraction and tracking: Task, evaluation, techniques and challenges, in: Proceedings of the International Conference RANLP-2009, Association for Computational Linguistics, 2009, pp. 166–172.

[7] M. Palmer, D. Gildea, P. Kingsbury, The Proposition Bank: An Annotated Corpus of Semantic Roles, Computational Linguistics 31 (2005).

[8] A.-L. Minard, M. Speranza, E. Agirre, I. Aldabe, M. van Erp, B. Magnini, G. Rigau, R. Urizar, Semeval-2015 task 4: Timeline: Cross-document event ordering, in: Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval '15, Association for Computational Linguistics, 2015, pp. 778–786.

[9] Semeval-2015, International Workshop on Semantic Evaluation, 2015.

[10] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60.

[11] T. Caselli, A. Fokkens, R. Morante, P. Vossen, SPINOZA_VU: An NLP Pipeline for Cross Document TimeLines, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 787–791.

[12] B. Moulahi, J. Strötgen, M. Gertz, L. Tamine, HeidelToul: A Baseline Approach for Cross-document Event Ordering, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 825–829.

[13] B. Navarro, E. Saquete, Gplsiua: Combining temporal information and topic modeling for cross-document event ordering, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 820–824.

[14] H. Llorens, E. Saquete, B. Navarro-Colorado, Applying Semantic Knowledge to the Automatic Processing of Temporal Expressions and Events in Natural Language, Information Processing & Management 49 (2013) 179–197.

[15] H. Llorens, E. Saquete, B. Navarro-Colorado, Automatic System for Identifying and Categorizing Temporal Relations in Natural Language, International Journal of Intelligent Systems 27 (2012) 680–703.

[16] E. Laparra, R. Agerri, I. Aldabe, G. Rigau, Multilingual and cross-lingual timeline extraction, CoRR abs/1702.00700 (2017).

[17] E. Reiter, R. Dale, Building Natural Language Generation Systems, Cambridge University Press, New York, NY, USA, 2000.

[18] A. Gatt, E. Reiter, Simplenlg: A realisation engine for practical applications, in: Proceedings of the 12th European Workshop on Natural Language Generation, ENLG '09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 90–93.

[19] D. A. Smith, H. Lieberman, Generating and interpreting referring expressions as belief state planning and plan recognition, in: A. Gatt, H. Saggion (Eds.), ENLG 2013 - Proceedings of the 14th European Workshop on Natural Language Generation, August 8-9, 2013, Sofia, Bulgaria, The Association for Computer Linguistics, 2013, pp. 61–71.

[20] D. Duma, E. Klein, Generating natural language from linked data: Unsupervised template extraction, in: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers, Association for Computational Linguistics, Potsdam, Germany, 2013, pp. 83–94.

[21] I. Konstas, M. Lapata, Inducing document plans for concept-to-text generation, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 1503–1514.

[22] R. Kondadadi, B. Howald, F. Schilder, A statistical nlg framework for aggregated planning and realization, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 1406–1415.

[23] M. E. Vicente, C. Barros, E. Lloret, Statistical language modelling for automatic story generation, Journal of Intelligent and Fuzzy Systems 34 (2018) 3069–3079.

[24] D. Dannélls, Multilingual text generation from structured formal representations., University of Gothenburg, Göteborg, 2012.

[25] C. D. Manning, H. Schütze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA, USA, 1999.

[26] A. Mnih, Y. W. Teh, A fast and simple algorithm for training neural probabilistic language models, in: Proceedings of the 29th Interna-

tional Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012.

[27] M. Ballesteros, B. Bohnet, S. Mille, L. Wanner, Data-driven sentence generation with non-isomorphic trees, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 387–397.

[28] A. See, P. J. Liu, C. D. Manning, Get to the point: Summarization with pointer-generator networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2017, pp. 1073–1083.

[29] Y.-C. Chen, M. Bansal, Fast abstractive summarization with reinforce-selected sentence rewriting, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2018, pp. 675–686.

[30] J. Cordeiro, G. Dias, P. Brazdil, Rule induction for sentence reduction, in: L. Correia, L. P. Reis, J. Cascalho (Eds.), Progress in Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 528–539.

[31] M. Valizadeh, P. Brazdil, Exploring actor–object relationships for query-focused multi-document summarization, Soft Computing 19 (2015) 3109–3121.

[32] N. Chambers, D. Jurafsky, Unsupervised learning of narrative event chains, in: K. R. McKeown, J. D. Moore, S. Teufel, J. Allan, S. Furui (Eds.), ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA, The Association for Computer Linguistics, 2008, pp. 789–797.

[33] N. Chambers, D. Jurafsky, Unsupervised learning of narrative schemas and their participants, in: K. Su, J. Su, J. Wiebe (Eds.), ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural

Language Processing of the AFNLP, 2-7 August 2009, Singapore, The Association for Computer Linguistics, 2009, pp. 602–610.

[34] J. C. K. Cheung, H. Poon, L. Vanderwende, Probabilistic Frame Induction (2013).

[35] N. Chambers, Event Schema Induction with a Probabilistic Entity-Driven Model, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013) (2013) 1797–1807.

[36] N. Mostafazadeh, From Event to Story Understanding, Ph.D. thesis, University of Rochester, 2017.

[37] K. Markert, S. Martschat, Improving ROUGE for timeline summarization, in: M. Lapata, P. Blunsom, A. Koller (Eds.), Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers, Association for Computational Linguistics, 2017, pp. 285–290.

[38] J. Allan, R. Gupta, V. Khandelwal, Temporal summaries of news topics, in: W. B. Croft, D. J. Harper, D. H. Kraft, J. Zobel (Eds.), SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA, ACM, 2001, pp. 10–18.

[39] H. L. Chieu, Y. K. Lee, 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, ACM, New York, NY, USA, 2004, pp. 425–432.

[40] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, Y. Zhang, Evolutionary timeline summarization: A balanced optimization framework via iterative substitution, in: In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR â11, ACM, 2011, pp. 745–754.

[41] G. Tran, M. Alrifai, E. Herder, Timeline summarization from relevant headlines, in: A. Hanbury, G. Kazai, A. Rauber, N. Fuhr (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham, 2015, pp. 245–256.

[42] W. Y. Wang, Y. Mehdad, D. R. Radev, A. Stent, 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, Association for Computational Linguistics, 2016, pp. 58–68.

[43] G. B. Tran, A. T. Tran, N.-K. Tran, M. Alrifai, N. Kanhabua, Leverage learning to rank in an optimization framework for timeline summarization, SIGIR 2013 Workshop on Time-aware Information Access (TAIA'2013) (2013).

[44] G. Binh Tran, M. Alrifai, D. Quoc Nguyen, Predicting relevant news events for timeline summaries, in: Proceedings of the 22Nd International Conference on World Wide Web, WWW '13 Companion, ACM, New York, NY, USA, 2013, pp. 91–92.

[45] B. Navarro-Colorado, E. Saquete, Cross-document event ordering through temporal, lexical and distributional knowledge, Knowl.-Based Syst. 110 (2016) 244–254.

[46] A.-L. Minard, M. Speranza, R. Urizar, B. Altuna, M. van Erp, A. Schoen, C. van Son, Meantime, the newsreader multilingual event and time corpus, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016).

[47] R. Saurí, J. Littman, R. Knippen, R. Gaizauskas, A. Setzer, J. Pustejovsky, TimeML Annotation Guidelines 1.2.1 (http://www.timeml.org/), 2006.

[48] J. Pustejovsky, J. M. Castaño, R. Ingria, R. Saurí, R. J. Gaizauskas, A. Setzer, G. Katz, D. R. Radev, Timeml: Robust specification of event and temporal expressions in text, in: M. T. Maybury (Ed.), New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA, AAAI Press, 2003, pp. 28–34.

[49] I. T. W. Group, ISO TimeML TC37 draft international standard DIS 24617-1, 2008.

[50] C. A. Bejan, S. Harabagiu, Unsupervised Event Coreference Resolution, Computational Linguistics 40 (2014) 311–347.

[51] J. R. Firth, Papers in Linguistics (1934-1951), Oxford University Press, Oxford, 1957.

[52] Z. Harris, Mathematical structures of language, Wiley, New York, 1968.

[53] P. Turney, P. Pantel, From frequency to meaning: Vector space models of semantics, Journal of Artificial Intelligence Research 37 (2010) 141–188.

[54] P. Gärdenfors, The geometry of meaning. Semantics based on conceptual spaces., MIT Press, Cambridge, Mass., 2014.

[55] L. Padró, E. Stanilovsky, FreeLing 3.0: Towards Wider Multilinguality, in: Proceedings of the Language Resources and Evaluation Conference (LREC 2012), ELRA, Istanbul, Turkey.

[56] J. Mitchell, M. Lapata, Composition in Distributional Models of Semantics, Cognitive Science 34 (2010) 1388–1429.

[57] J. A. Bilmes, K. Kirchhoff, Factored language models and generalized parallel backoff, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003–short Papers - Volume 2, pp. 4–6.

[58] A. Stolcke, Srilm – an extensible language modeling toolkit, in: IN PROCEEDINGS OF THE 7TH INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING (ICSLP 2002, pp. 901–904.

[59] K. K. Schuler, Verbnet: A Broad-coverage, Comprehensive Verb Lexicon, Ph.D. thesis, 2005.

[60] C. Fellbaum, WordNet: An Electronic Lexical Database., MIT Press, 1998.

[61] A. Isard, C. Brockmann, J. Oberlander, Individuality and alignment in generated dialogues, in: Proceedings of the INLG, Association for Computational Linguistics, 2006, pp. 25–32.

[62] E. Lloret, M. Palomar, COMPENDIUM: a text summarisation tool for generating summaries of multiple purposes, domains, and genres, Natural Language Engineering 19 (2013) 147–186.

[63] A. F. Sevilla, A. Fernandez-Isabel, A. Díaz, Enriched semantic graphs for extractive text summarization, in: Conference of the Spanish Association for Artificial Intelligence, Springer International Publishing, Springer International Publishing, 2016.

[64] F. Andonov, V. Slavova, G. Petrov, On the open text summarizer, International Journal "Information Content and Processing" 3 (2016) 278–287.

[65] C.-Y. Lin, ROUGE: A Package for Automatic Evaluation of Summaries, in: Text Summarization Branches Out: Proceedings of the Association for Computational Linguistics Workshop, Association for Computational Linguistics, 2004, pp. 74–81.

[66] C.-Y. Lin, E. Hovy, Automatic evaluation of summaries using n-gram co-occurrence statistics, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, 2003, pp. 71–78.

[67] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Çelebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, Z. Zhang, Mead - a platform for multidocument multilingual text summarization, in: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), European Language Resources Association (ELRA), 2004.