Semantic Patterns to Structure Timeframes for Event Ordering Enhancement

Nour Matta
IT and Digital Society Laboratory
University of Technology of Troyes
Troyes, France
email: Nour.matta@utt.fr

Nada Matta
IT and Digital Society Laboratory
University of Technology of Troyes
Troyes, France
email: Nada.matta@utt.fr

Nicolas Declercq
Namkin
Troyes, France
email: n.declercq@namkin.fr

Agata Marcante
Research & Development
Namkin
Troyes, France
email: a.marcante@namkin.fr

***Abstract*— Event ordering is a field in Event Extraction that deals with the temporality aspect and order of occurrences of events mentioned in a text. Event Ordering is essential because any analysis of causalities and consequences of specific actions or changes of state requires a time evaluation. Standard approaches using machine learning models, with or without inferences, start by identifying events in text and then generate the temporal relationships between them individually. With no consideration of flashbacks, flash-forward, and direct speech temporal aspect, available models lack performance. In this paper, we introduce a novel approach to group events in temporal frames that we refer to as Timeframes. Three types of timeframes will be presented: Publication, Narrative, and Spoken. The purpose of this paper is to highlight the need of this approach, define the different timeframes, introduce their extraction process, evaluate the extraction and compare the event ordering with and without the timeframe approach.**

Keywords-Timeframe; Event Extraction; Event Ordering; Natural Language Processing.

I. INTRODUCTION

Event extraction is one of the most important tasks of Information Extraction through Natural Language Processing [1], [2]. It enables the extraction of events in text and aims to identify the different participants and attributes of the extracted events. Some examples of the extracted information can be the cause, place, time, means, or goal, which can be identified through dependency analysis [3]. Moreover, evaluating the influence of a particular event or a specific action requires an account of temporality [4]. In traditional event extraction, available approaches are very performant when it comes to the analysis of single sentences. Some approaches can support complex sentences. But even though models aim to extract events from a "text" and create temporal relations between them, the performance lacks, and soon the extracted information easily becomes unreadable or inaccurate from a temporal relation point of view [5]. Furthermore, when focusing on temporality event extraction, many approaches focused on ordering the multiple events mentioned one by one and creating relationships [6], [7] without considering the fact that multiple 'processes' can be part of a preparatory stage of a single event. In this paper, we introduce the use of timeframes, an approach used for time analysis in different domains, to improve the temporal relation made between events in a text.

It is important to note that within the same text, multiple timeframes can be identified and multiple time references can be used. A small example would be a news report about a company announcing the launching of a new product. We have the time when the announcement was made, the timeframe within the announcement (such as the date of the launching), and the time of the publication of the news. Another example would be in a narrative text in which the author talks about multiple events while going back and forth in time. Our main goal is to identify the events in a text, create temporal relations between them and identify the different timeframes if there are multiple ones. We aim to assign each event to its timeframe enabling improved readability of the extracted event, their temporal relation, and finally their interpretation.

In Section 2, we will start by defining what timeframes are and how they are used for time analysis. We will also present the different conceptualizations of events in linguistics. In Section 3, the related work, we will go through the different event extraction approaches. In Section 3, we will be presenting the timeframe approach along with part of the semantic pattern identified. In Section 4, we will provide an example of application of the proposed algorithm. In Section 5, the result and evaluation done on 120 news article from multiple source and multiple topic. Lastly, Section 5 provides the future work.

II. TIMEFRAMES AND EVENTS

This section is divided into two main parts: the timeframes and the events. For the timeframe, we will go through the analysis of temporality in fields other than text mining and show how those conceptualizations can be helpful in the analysis of temporality in text. As for the events, we will go through their definition in the event extraction field and how it is viewed from a linguistic point of view.

A. Timeframes

A timeframe is a certain period of time in which an event should happen or has already happened [8]. This leads us to question the meaning of time. In philosophy [9], the Platonist understanding of time is segregated from the Relationist definition. Platonists picture time as an "empty container" of events that exist regardless of whether anything is placed in it. In this perspective, Platonists consider that it is possible that changes in the universe can cease to exist for a certain period. On the other hand, Relationists view time as a set of events and the temporal relationship between these events. While dealing with event extraction and their temporal relationship, the Relationist understanding is used.

The study of the temporal relation between actions, events, states, and their influences is applicable in different domains other than event extraction. In their study on temporality in video games, Zagal et al. [4] distinguished multiple types of games: the ones with game time being equivalent to the real-world time, the ones in which action can speed up or skip time, the ones where specific action triggers events of a specific duration, and finally the ones where certain events occur without affecting the game time as if time had stopped. To analyze the temporality for each game type, Zagal et al. [4] defined timeframes, creating relations between those timeframes and between events within the timeframe and coordinating them. Reflecting on that approach, from a textual perspective, the authors also set the duration to specific events as shown in "1)", which can make flashbacks "2)" and flash-forwards "3)". They can also skip time "4)" and even focus on a specific event or describe elements making the time indirectly stop "5)".

1) *John ran for an hour.*
2) *Henry was looking at the photo. He took it a few years back when he was in New York.*
3) *John is preparing his luggage; he will be leaving in the morning.*
4) *Five years later, Henry went back to New York.*
5) *John looked through the window for a few seconds. It was a rainy day; people were walking while holding their umbrellas. He went to his desk.*

Distinguishing the different timeframes and specifying the events that happened in each frame enables the focus on specific events based on their occurrence time and aims to improve coordination between multiple timeframes in the text. However, in the event extraction field, events are ordered one by one without having a more global representation. Some of the concepts that must be considered while dealing with temporality are the duration, the time point, calendar, narrative time, timeline, countdown, and temporal relation [4], [8]. Each of these concepts plays a specific role in the pattern and the extracted knowledge. The use of timeframe also enables the consideration of the release date of the text as a timeframe on its own in order to improve topic tracking and event follow-up.

B. Events

In the event extraction field and the event-based decision systems, events are usually defined as happenings or changes that occurred in a specific interval of time. They can be associated with the change of states (canceled, ongoing, recently done, past or future plans) and can have multiple occurrences [2], [10].

Other than Natural Language Processing, linguists also worked on defining what an event is, distinguishing it from a state, and partitioning it onto atomic and extended events. Using the tense of the verb, the duration, and time reference along with temporal connectors, some set of rules and patterns are proposed. Vendler [11] was one of the first to work on defining the concept of event in linguistics, while working on verbs and tenses, he first identified the tense as the location of a happening in the time (past, present, or future) and its aspect which refers to the state of an event (completed, ongoing or interrupted). Later on, he defined "Eventualities" [11] as a concept that groups the state and non-state. Some particularities of each group were identified:

6) *Jack was ill on Sunday*.
7) *Jack wrote a letter on Sunday*.

"6)" is an example of a state, in which we cannot determine if the state "ill" started before or during the "Sunday" and if is stopped during "Sunday" or after. While the non-state "wrote" started on ended Sunday. And comparing the duration of "was ill" and wrote, we can presume that "wrote" has a shorter duration than "was ill".

Their conceptualization of eventualities goes as follows: non-state was divided into Activities and Events; activity refers to actions that had a duration but with no endpoint or consequent state while events have a quantification or an ending result:

8) *Alex ran.*
9) *Alex ran to the store.*
10) *Alex ran a mile.*

"8)" is considered an activity while "9)" and "10)" are events. Events are then distinguished [12]. Where an accomplishment is considered to have a duration and accept the progressive (continuous tense) while achievement is strange in progressiveness. It is important to note that Kamp highlighted the ambiguity between those concepts, starting with the very first division between distinguishing a state and a none state.

Using the conceptualization made by Vendler, Moens et al. [10] defined another conceptualization. Eventualities are divided into States and Events. And they considered two dimensions for distinguishing events: the duration and the consequence. For the duration, they considered events as Atomic or Extended. Extended Events have a notion of duration. For the consequence, they started by defining the term "culmination" as an event that has a consequence, a change of state. A "nucleus", as shown in Figure 1, is the combination of a preparatory process of a culmination, the culmination, and the consequent state. If we consider the example "9)", "Alex ran to the store", we can regard it as a culmination, in which running is a preparatory process to the culmination of arriving at the store. The consequent state is "being in the store".

Figure 1. Moen et al. nulcreus definition [10].

Figure 2 presents the 4 subcategories of events. An Atomic Event with no consequence is considered as a point, for instance, "He hiccupped". A culmination is an atomic event with a consequence [10]. An extended event is a process and is considered a culminated process if it has a consequence. It is important to highlight that many elements were used to distinguish the different categories of events. This work and the pattern identified in it are essential for our approach especially the use of a nucleus. When using the timeframe approach, identifying the culminations in a text and all the processes, the preparatory stage, and the consequent state are one of our goals. The slice difference in our approach is that we intend to associate different events on the nucleus timeline.

	Atomic Event	Extended Event
- Consequence	Point	Process
+ Consequence	Culmination	Culminated Process

Figure 2. Moens' event conceptualization [10].

The pattern identified by Moens will be used and associated with other patterns to enable the representation of events extracted for each timeframe. When trying to identify events and states, the use of adverbs, the tense of the verb, and the use of semantic dependencies, such as the verbs' objects, were used for identifying the categories. The same verb can be considered a point, a process, a culmination of a culminated process depending on its use.

It is important to note that in linguistics, verbs tend to be classified as states and events. Adjectives are considered states of the elements they describe. In the event extraction field, nouns are also identified as events depending on their context, for example:

11) *Two years after his graduation, John moved to New York.*

In this sentence, the noun "graduation" is considered as an event. In order to manage these types of events, a pattern concerning nouns was added. If a noun is a temporal reference, in this case, "after graduation", then this noun is an event. Several studies define principles to identify and extract events along with their arguments; we summarized them in the related work.

III. Related Work

This section will be partitioned as follows: we will start by going through event extraction techniques and more importantly Event Ordering. Then we will go through different research works that addressed temporality aspects in the temporality recognition techniques.

A. *Event Extraction*

Event Extraction is one of the most crucial branches of Information Extraction from textual data. In this field, an event is presented as an occurrence of a happening [5], [10]. Events may be related to a specific date, actors, places, objects, or other events. Event Extraction models provide effective benefits to build decision support systems or question-answering models [13]. The first work on event extraction started in the health care field in order to analyze bacterial behavior [14], and thanks to the progressive advancement of data science and big data, this field gained popularity to enable insights extraction from the published or shared textual data. Some of those data sources are social media posts, news messages, web pages, articles, or research papers. Knowledge discovery strategies that are event-oriented start with event identifications along with the related entities that play a role in those events and enable decision models based on the extracted information [15], [16].

Event extraction is divided into two types [7]: Open-domain event extraction and Closed-domain event extraction. Open-domain event extraction is a non-specific event detection and extraction. This approach is independent of annotated data. The detection of events and their arguments is based on their syntactic and semantic role. The arguments types are not specific to the event type but rather standard arguments such as actor or agent, location, object, time, cause, etc. This method is used for topic detection, story segmentation, and first story detection. Open-domain event extraction is also referred to as a data-driven approach [5] where researchers aim to convert data into knowledge using statistical methods and data mining. On the other hand, closed-domain event extraction is the extraction of previously defined events of interest. Those events are relative to the application domain. For instance, in the finance sector, business intelligence decision support models require economic event analysis. Event extraction enables the detection of business events along with the entities involved in each event [17]. An example could be a merger of two companies, the models aim to distinguish the old companies that had emerged from the newly created company. Another scenario can be seen in police reports analysis. Homicide or accident incidents may be detected along with specific arguments for each type of event that are precisely named and pertinent to certain occurrences including the location, the period, the victims, the offenders, etc. This type of event extraction can be used for event trigger detection, event mention detection, event argument extraction, and causal relation event analysis. There exist multiple methods that can be used for closed-domain event extraction. The pipeline approach [18] is a popular method that divides the extraction into multiple classification stages. The process is as follows: first, it uses keyword matching to identify the event trigger. Second, arguments are classified based on semantic and syntactic dependencies. Arguments can be simple words, noun phrase segments, or verb phrase segments. Lastly, a role is assigned for each argument. Another popular approach is the join-based approach in which the events type and the arguments are simultaneously classified. The purpose of this simultaneous classification is to avoid errors generated by the trigger classification set. To improve argument role identification, the reinforcement learning approach and the incremental learning model [6] can be applied. These methods distinguish different arguments related to an event within the same sentence. Deep-learning methods [19] train their models based on annotated datasets and use transformers such as BERT. Their performances depend on the quantity and quality of the annotations taken as input.

Knowledge Graphs based on event extraction are one of the potential representations for the events and their related entity [20]. Two types of representation are popular: event-centric and entity-centric knowledge graphs. Event-centric uses only events as nodes and the entities connected to the events are added inside the nodes as attributes for better visualization of all the events and how events are related to one another. On the other hand, entity centric considers all entities as nodes. This view enables a detailed representation of each event. In our work, we focus on events and only the temporal relations between them. In other words, we will be using the event-centric representation to represent the events and their relations in the provided figures.

B. *Event Ordering*

In order to study causal, consequential, or impart effects of events and happenings, time analysis of events occurring is essential [21]. Temporal relations analysis is a key factor for narrative processing, storyline construction, causality, and impact evaluation. In the question-answering field, temporal analysis was used to evaluate the correctness of the answer based on the temporal aspect. Two main categories of questions were identified. The first set of questions had permanent answers or answers for a long period of time such as "Who was the US president in 1996". The second set had answers that were short-term and change a lot over time such as: "Who is the US president today" [13].

Two main temporal event relation extraction approaches are available: the data-driven approach and the hybrid approach which uses both data-based and constraint-driven approaches [22]. Data-driven models are trained using annotated datasets, and the most method for event ordering is the join-based approach. The constraint-based approach [23] enables applying inferences while processing the data to generate more information. An example of inferences is the transitivity of the event order or the reverse relation generation. For example, if event A happened before event B, the relation event B happened before event A can be created. Another example would be considering the three events A, B, and C, if A happened before B and B happened before C, then A happened before C.

There exist multiple datasets for temporal relation extraction, different versions of them, and those datasets differ in the relationships available, the type of entities linked, and when the relationship is considered or ignored. An example of the dataset would be TCR [24] which considers only five types of temporal relationships: "before", "after", "is included", "includes", and "simultaneously". TimeBank [25] on the other hand also considers "ended by", "During", "Begun by", "Begins", "IBefore", "IAfter", "During", and "Ends". While TimeBank Dense [26] has a relation named "vague" that relates events with unclear temporal relations. Some datasets have annotation features of the event aspects label. In TimeML [27] for example, "Progressive", "Perfective", "Perfective Progressive" or none are assigned to each event.

There are three type of temporal relations available [28]: (1) relations between two events, (2) relations between events and time expressions, and (3) finally relations between time expressions and time expressions. Time expressions can be labeled as duration, date, time, or set. To compute the time value, TimeN provides a time normalization system that converts time expressions to actual Date Time relations [29]. It's important to note that not all models consider all three types of relationships. Events can be nouns and verbs, the models match the tense of the verbs but don't consider the culmination, point, process, and culmination process aspects. Events are assigned polarities such as negative and positive and modularity such as "would", "may", or "could".

Multiple models are presented such as NavyTime [28], ClearTK [27], UTTime [30], CAVEO [26], Sequential Models based on LSTM [31], TEKMN [13], or structured learning approach [21]. Temporal label dependencies and constraints are used to improve relations between events [27]. Some worked on the linguistic and syntactic rules, such as Leeuwenberg et al. [23] or Laokulrat et al. [30]. The models vary in the search process, especially in terms of the distance between the events and time expression. Some models search for temporal relation intra-sentences; others go beyond single sentences. For instance, NavyTime searches for temporal relations of events are not only limited to the same sentence, but also in adjacent sentences, and between paragraphs. Another example would be UTTime which considers the relations between all events and the document creation time, events and time expression, and events mentioned in the same sentence and in consecutive sentences. The performance of the models depends on the datasets used along with the temporal relation types, and the overall search process, some models are built using multiple datasets.

The models that uses inference may go not only be limited to temporal relationship in their search. Since a cause occurs before its consequences, causal analysis enables temporal inferences [24]. So other than the previously mentioned transitivity and reverse inferences, the causal relation generates an ordering inference between the event and its cause. And vice versa, some models work on both temporal and causal relations and add constraints to learn more about the causal relations based on event ordering. Some of the datasets that used causal relations are Causal-TimeBank [32], Event Storyline [33], and FinReason [34].

Lastly, TIMERS [35] is considered a document-level approach that integrates causal prerequisite links, chain reasoning, and future events to improve temporal relationship extraction.

In this paper, we introduced timeframes as frames of time that will contain multiple events that happened in a specific period of time. This approach aims to group events together by detecting relationships between the different timeframes identified even though they are not mentioned in adjacent sentences. After detecting the relationship, we will apply one of the related work approaches for the event order. In this paper, we will be using the CAEVO model based on the TimeBank-Dense dataset for its availability. In the next section, we will present the different types of timeframes identified, their use, and how to extract them.

C. Temporality Recognition Techniques

In the Question Answering field, temporal analysis is a must for determining if an answer to a question will change throughout time or not. In their work, Pal et al. [36] identified multiple classes of information temporality: short duration, medium duration, long duration, and permanent. They tried classifying the question/answer under those categories but ended up grouping the short term and medium term together and long term and permanent together. It enabled distinguishing between “Who won the competition X in 2022?” and “Who won the last competition?”. One of the questions will have permanent information and the other will change throughout the years. It is important to consider these types of classifications to identify information that is true regardless of the timeframe of the text and relations that are relative to the timeframe of the text. Recent work focuses on identifying the attention in complex questions and the use of multiple sentences that contain the answer [13]. Note that in their work Kwiatkowski et al. [37] mentioned descriptive sentences or informative sentences, in which information is given without a particular event being mentioned.

Temporality plays a very important part in social science and social discourse analysis [38]. Coordination between different events from multiple resources is also used when clustering news and following up on events. Sources vary between news and social media posts, such as tweets [39]. It is also essential to consider time relations when analyzing the influence of social media and the media in general on social events, such as protests and violence and study the sentiments behind it [40].

The question answering field provided a very important aspect to consider when extracting events and information. Completed Events and states with a specific date tend to be permanent information while unfinished events and events with reference to the text temporality tend to be true in a specific timeframe. Coordination of events between multiple texts will be considered in our approach and will be based on the timeframe concept. Our approach introduces the use of multiple types of timeframes and how to extract them. We will be using several models and patterns already provided in order to optimize the model’s performance.

IV. Timeframe Approach

For the extraction of timeframes that will be used to improve the temporal relation analysis between events, we identified three types of timeframes, (1) the Publication Timeframe, (2) the Narrative Timeframe, and (3) the Spoken Timeframe. Those timeframes were inspired by the identified timeframes for temporal analysis in video games with adaptation to the text constraints [4]. The Publication Timeframe reflects the publication date or year of the analyzed text. The Narrative Timeframe is the timeframe of the events happening in the text; we may find multiple Narrative Timeframes in a single document. Finally, the Spoken Timeframe is a particular type of timeframe that may not always appear in a text. It is used when an announcement, a speech, or a dialogue is present. The events and information that are mentioned in that context will be analyzed in their own timeframe in order to reduce event relationship complexity. The timeframe will consist of two main parts: (1) the text belonging to the timeframe, and (2) the extracted information related to it.

A. Publication Timeframe Extraction

All text documents have by default a Publication Timeframe and a Narrative Timeframe. To identify the publication date, the type of text affects the extraction. If a post on social media is being analyzed then, the date is usually available as metadata to the text. When dealing with online news, most publishers put the date at the beginning of the text. Considering the presence of the title, we will check the first three sentences, for the presence of dates using Named Entity Recognition. If no dates were found, the last sentence will be checked. In case a sentence was identified as the publication date, it will be extracted from the document in order to avoid confusion with the rest of the text. The date will be set in the information field of the timeframe. Figure 3 provides the Publication Timeframe extraction function. It takes two elements as input: a text, and the patterns that identify the publication date. The returned list contains two elements: The Publication Timeframe and the text. The text is returned since it is modified in case the pattern was found in a sentence.

```
Extraction_Publication(Single_Text, pub_pattern):
    sent = split_sentences(single_Text)
    tf_pub = [[],[]]
    to_check = [sent[0],sent[1],sent[2],sent[-1]]
    for element in to_check:
        identified = check_pattern(pub_pattern,element)
        if identified:
            timeframe_pub = [[date],[element]]
            remove element from Single_Text
            return [timeframe_pub , Single_Text]
    return [[], Single_Text]
```

Figure 3. Publication Timeframe Extraction.

Table 1 provides some of the patterns used to identify the Publication Timeframe. Please note that for the first two patterns, their presence in the sentence is enough while for the last two, they must be alone in the sentence to be considered a sign of Publication Timeframe.

TABLE I. SOME OF THE PATTERN USED TO DETECT PUBLICATION TIMEFRAMES

pub_patterns
'Updated' + <date>
'Published' + <date>
<number> + [hours, days, months, years] + 'ago'
<date>

When identifying temporal relations between timeframes, the publication timeframe will have relations with the narrative timeframe. This will enable possible relation of the events mentioned in the narrative timeframes and the publication date.

B. Spoken Timeframe Extraction

```
Extraction_Spoken(Single_Text, say_pattern):
    list_sentence = split_in_sentences(single_Text)
    tf_speech_element = [[],[]]
    tfs_speech =[]
    id = 0
    before = false
    for sentence in list_sentence:
        identified = check say_pattern in sentence
        if identified and not before:
            before = true
            tf_speech_element = [id,[sentence]]
            replace(sentence,Single_Text,"tf_speech_"+id)
        else:
            if identified and before:
                add sentence to tf_speech_element[2]
                remove sentence from Single_Text
            else:
                if before:
                    before = false
                    add tf_speech_element to tfs_speech
                    id = id + 1
                    tfs_speech = [[],[]]
    return [tfs_speech , Single_Text]
```

Figure 4. Spoken Timeframe Extraction.

This timeframe will be treated before the Narrative Time. The search for verbs that reflect speaking and punctuation that are proper to dialogue will be the main task. If nothing is identified, we skip to the next stage, else a spoken frame will be created. If the "spoken" elements are all available in successive sentences, they will all be extracted and set in a single Spoken Timeframe. If multiple sentences have 'spoken' elements but are not successive, a Spoken Timeframe should be created for each nonconsecutive part. But in order to enable relations between the Narrative Timeframe and the Spoken Timeframe, identification will be assigned to each extracted Spoken Timeframe, and the extracted sentences will be replaced by the Spoken Timeframe Identification. If any dates are mentioned, they can be added to the information field of the timeframe. Note the tense in the Spoken Timeframe reflects a relationship between the Spoken and Narrative Timeframe it belongs to, so if a unique tense is identified, a relation between the Spoken and the Narrative Timeframe will be identified. For example, if future tense is identified in the "spoken" element, then the relationship will most probably be "after". To keep track of this relationship, the relation if available will be added with the timeframe identification.

Figure 4 provides the Spoken Timeframe extraction function. It takes two elements as input: a text, and the patterns that identify the speaking patterns. The returned list contains two elements: The Spoken Timeframe list and the text. The Spoken Timeframe list contains all the Spoken Timeframes identified in the text. Each one contains an identification that distinguishes different segments in which the patterns were identified along with the sentences. Note that if consecutive sentences contain the patterns, then they will be grouped in the same timeframe. The return text is the remaining text with the identifications of the Spoken Timeframes.

We used a single pattern to identify the presence of a direct speech in a sentence. First, some direct speech may contain multiple sentences between quotation mark which reduced the accuracy of the dependency parsing. This is why, during the pattern matching phase, any text between quotation was replaced by "" and we analyzed the dependencies of the quotations. If the quotation mark is the object of the verb in the sentence, then we consider that the current sentence belongs to a Spoken Timeframe.

A small modification was brought to our algorithm to enable the next step of this approach that differs from Matta et al. [1]. In order to evaluate the temporal relation between the different timeframes, we should keep track of the tense of the verbs available in the direct speech. Therefore, we applied the algorithm of verb tense extraction. The tense extraction algorithm is presented in section C. The tense in the direct speech reflects the relations that the spoken timeframe has with the narrative timeframe it belongs to. For instance, if the tense is in the past such as "'He worked on the program yesterday', Simon said.", then the spoke timeframe reflects on events that occurred before the narrative timeframe they were mentioned in.

C. Narrative Timeframe Extraction

The starting point of the Narrative Timeframe is having an empty information field and the whole text inside of it. The purpose of using multiple timeframes is to distinguish between current time in a text and in case a flashback is mentioned or flash-forward is mentioned, the information should be treated accordingly. Using the VerbNet parser [41], we detect any temporal relation. We associate a change in the timeframe when the relationship is not related to a specific event. For example, "before going to bed" is related to the event "go to bed" while "a few years ago" is a temporal relation with the current timeframe. We also consider "later that day" or "later that year" elements within the same timeframe.

In this section, we will present the elements that trigger the creation of a new Narrative Timeframe. The temporal relationship elements that will create a new Narrative Timeframe are: "a few years later", "(number) years later". The same goes for "months" and "days" instead of "years" and "ago" instead of "later". Dates are relatively important; if a date is mentioned, it will be assigned as information about the timeframe. If no dates are mentioned, temporal relations that start with ‘this’ for example, ‘this year’, ‘this month’, ‘today’, will be considered as time information of the timeframe. If multiple dates are separately mentioned, each will be assigned a timeframe.

```
Extraction_Narrative(Single_Text, narrative_pattern, tense_pattens):
    list_sentence = split_in_sentences(single_Text)
    tf_nar_element = ["tf_nar_0",[],[]]
    tfs_narrative = []
    id = 0
    tocheck = 0
    for sentence in list_sentence:
        identified = check narrative_pattern in sentence
        if tocheck == 0:
            if not identified :
                add sentence to tf_nar_element[1]
            else:
                dominant_tense = check_tense(tf_nar_element)
                id=id+1
                add "tf_nar_"+id to tf_nar_element[1]
                add tf_nar_element to tfs_narrative
                sentence_tense = check_tense(sentence,tense_pattens)
                tf_nar_element = ["tf_nar_"+id,[sentence],[sentence_tense]]
                if (dominant_tense != sentence_tense):
                    tocheck = 1
        else:
            if not identified :
                sentence_tense = check_tense(sentence)
                if sentence_tense == tf_nar_element[2]:
                    add sentence to tf_nar_element[1]
                else:
                    add tf_nar_element to tfs_narrative
                    id=id+1
                    sentence_tense = check_tense(sentence)
                    tf_nar_element = ["tf_nar_"+id,[sentence],[sentence_tense]]
                    tocheck = 0
    add tf_nar_element to tfs_narrative
    return tfs_narrative
```

Figure 5. Narrative Timeframe Extraction.

Figure 5 provides the algorithm used for the Narrative Timeframes extraction. It takes as input the text, the patterns that identify the existence of a new timeframe, and the patterns that check the tense of a verb. The patterns that check the tense of the verbs are based on the part-of-speech tagging, dependencies, and the lemmatization of the verb. The lemmatization is the original form of a word without conjugation. We use it only to detect the verbs ‘be’ and “have”. Those elements are provided by Natural Language Processing tools such as Spacy [42]. For the part-of-speech tags of interest, we used:

- “VB” is assigned to the verbs base form
- “VBD” is assigned to verbs in the past tense
- “VBG” is the gerund (a verb that ends with ‘ing’)
- “VBN” assigned to the verb in past participle form
- “VBP” is assigned to the verbs in non-third person singular present form
- “VBZ” is assigned to the verbs in the third person singular present form

We considered the 12 principal tenses, and Table 2 provides some of the tenses and their respective patterns. We grouped the 12 verb tenses in the respective 5 tense categories: past anterior, past, present, future, future anterior [11]. For example, present continuous and present simple will both be present while present perfect, past simple, and past continuous will be considered as past. Based on the verb tense, the function check_tense will return the category of the verb tense identified.

As for the patterns that identify the presence of a new Narrative Timeframe, Table 3 presents some of them.

TABLE II. SOME OF THE PATTERN USED TO DISTINGUISH VERB TENSE

Verb Tense	tense_patterns
Present Simple	pos = “VBZ” or pos = “VBP”
Present Continuous	verb with pos=“VBG” and has_child = {dep= “AUX”, pos = “VBZ” or “VBP”, lemma=“be”}
Past Simple	verb with pos=“VBD”
Past Continuous	verb with pos=“VBG” and has_child = {dep= “AUX”, pos = “VBD”, lemma=“be”}
Future Simple	verb with pos=“VB” and has_child = {dep= “AUX”, pos = “MD”, lemma=“be”}

TABLE III. SOME OF THE PATTERNS THAT IDENTIFY NARRATIVE TIMEFRAMES

Narrative_Patterns
A few [‘years’, ’months’, ’days’] [‘later’, ‘ago’, ‘back’]
[‘earlier’, ’later’] [‘this’, ‘that’] [‘years’, ’months’, ’days’]
In <date>
[‘starting’, ‘from’, ‘starting from’] <date>
<number> [‘years’, ’months’, ’days’] [‘later’, ‘ago’, ‘back’]

The algorithm goes as follows: an empty Narrative Timeframe is initialized. We go through all the sentences and we check the presence of a pattern. If no pattern is identified, we add the sentence to the timeframe. If patterns that trigger the creation of a new Narrative Timeframe are identified, we generate an identification to the new timeframe and we add to the previous timeframe to keep track of their connection. We then check the tense of the previous timeframe and the tense of the new one and we save the current timeframe element in the list of Narrative Timeframes. If the tenses are similar, we just add the sentence to the new timeframe. If the tenses are similar, there is a high risk that the author switches back to the previous timeframe. In that case, for the upcoming sentences, we keep track of any changes in the tenses. This is the only case in which the change of tense will trigger a change in the Narrative Timeframe. In future work, just like the event ordering approaches, a change in tense will trigger relations between events. Examples (12) and (13) clarify the need for this process:

12) *John is thinking about his life in New York. A few years ago, he had to move out because of his parents’ job. He misses his friends dearly.*
13) *Alice graduated with a master's degree. A few months later, she found a job in an international company. She was finally able to move out.*

In 12), the change of the tense use can simulate a go back to the previous timeframe or just a need to change timeframes. In our current algorithm, we will just separate the three timeframes and handle the relationships between different timeframes in future works. As for 13), the continuity in the tense simulates just a skip in time with no need for further tense monitoring of verb tenses. In the next section, we will be evaluating our approach.

V. EXAMPLE OF APPLICATIONS

In this section, we will present an output for each of the three algorithms provided above. In this section, we will present the output for each of the three algorithms provided above. We used news articles to evaluate the performance of the algorithms. The articles differ in themes, sources, and sizes. The topics and the data sources used are:

- Sports from www.nba.com
- Automotive from www.autoweek.com
- Aeronautics from www.aerotime.aero
- Healthcare from www.healthcarenews.com
- Energy from www.euronews.com
- Politics from www.bbc.com, www.cnn.com, and www.glogalnews.ca

We aimed to have multiple styles in writing, and multiple sources. In politics, the main goal of the multiple sources was the evaluation of the publication timeframe extraction. We tool 20 news articles for each topic so in total we got 120 articles to evaluate. The results will be presented as follow: we will start by providing an example of the output of each of the algorithm alone then we will evaluate the overall output of all the data. Please note that the evaluation of the algorithms is manually done since this approach is still in its early stages and no automatic or pre-annotated data is available.

A. Publication Timeframe Extraction Results

For each site, we started by testing the performance of the Publication Timeframe since the scraper used is not customized for each website. We were able to identify the Publication Timeframe.

Figure 6. Output of Publication Timeframe Extraction Algorithm.

Please note that a cleaning phase is necessary before applying the approach. Figure 6 presents one of the outputs of the algorithms. The figure provides part of the extracted text from a news site with the sentence with the pattern of interest highlighted in the input of the algorithm. We can notice the Publication Timeframe along with the rest of the text from which we removed the sentence with the pattern.

B. Spoken Timeframe Extraction Results

Figure 7 provides the output of one of the provided data. Please note that for better visualization, long paragraphs with no pattern were replaced by '...' in the figure.

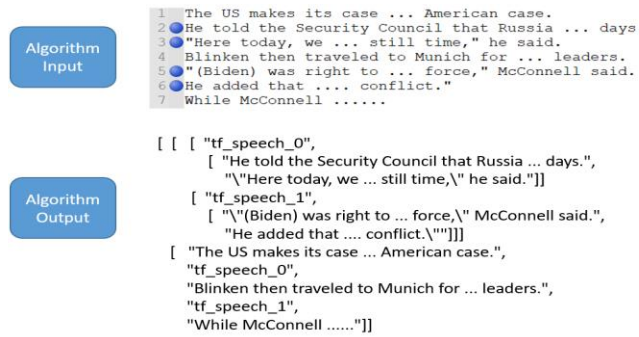


Figure 7. Output of Spoken Timeframe Extraction Algorithm.

In the example provided in Figure 7, we identified 4 sentences with patterns. They were distributed into 2 groups of consecutive sentences with patterns. The sentences in the example input were marked by blue dots next to them. In the output, we can notice that the algorithm provided a list with two elements in it, a Spoken Timeframe list and a list of the remaining sentences. The Spoken Timeframe list had 2 Spoken Timeframes in it, each having identification and consecutive sentences with patterns. As for the remaining text list, we notice that the extracted sentences were indeed replaced by their respective timeframe identification.

C. Narrative Timeframe Extraction Results

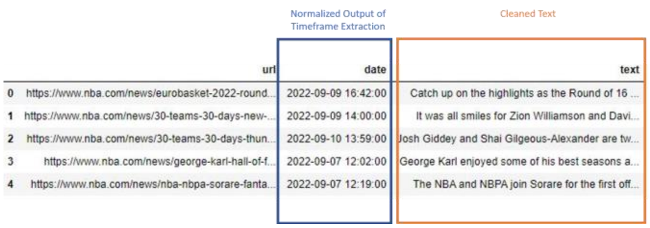
Finally, for the Narrative Timeframe, for the representation we used a text that had 2 Spoken Timeframes already identified in it. This enables an explicit viewing of how the identification and the linking between Narrative and Spoken is provided. The Spoken identifications in this example are "tf_speech_0" and "tf_speech_1". We also added blue dots next to the sentences with the patterns identified.

Figure 8. Output of Spoken Timeframe Extraction Algorithm.

We can notice in the output the Narrative Timeframe list returned by our algorithm in Figure 8. It contains the three expected timeframes having their respective identification, the sentences ordered that belong to the timeframe, and the tense of the last sentence to enable comparisons.

D. Results Analysis

The evaluation of the results will go as follow: we will tackle each timeframe type by order, Publication, Spoken, and Narrative. We started by separating the Publication Timeframe and we normalized the date available.



Normalized Output of Timeframe Extraction | Cleaned Text

	url	date	text
0	https://www.nba.com/news/eurobasket-2022-round...	2022-09-09 16:42:00	Catch up on the highlights as the Round of 16 ...
1	https://www.nba.com/news/30-teams-30-days-new-...	2022-09-09 14:00:00	It was all smiles for Zion Williamson and Davi...
2	https://www.nba.com/news/30-teams-30-days-thun...	2022-09-10 13:59:00	Josh Giddey and Shai Gilgeous-Alexander are tw...
3	https://www.nba.com/news/george-karl-hall-of-f...	2022-09-07 12:02:00	George Karl enjoyed some of his best seasons a...
4	https://www.nba.com/news/nba-nbpa-sorare-fanta...	2022-09-07 12:19:00	The NBA and NBPA join Sorare for the first off...

Figure 9. Output of Publication Timeframe Extraction Algorithm Normalized.

Figure 9 shows a segment of the output of the Publication timeframe after a Normalization of the available date. When the hour is not available it is by default assigned “00:00:00”. Once the data is cleaned we ran the Spoken Timeframe algorithm and finally the Narrative Timeframe algorithm. We started by evaluating the repartition of the Spoken Timeframes.

Figure 10. Pie Chart representing the repartition of the spoken timeframes.

Figure 10 provides the different repartitions of the of the spoken timeframes per domain. Sport and aeronautics both having around 25% with spoken timeframe, Automotive around 50% and energy, health and politics two third of the articles had Spoken Timeframes. The amount of data used is not sufficient to provide a deduction. It’s important to highlight the fact that the longer text with Spoken Timeframes are the more likely we find multiple Spoken Timeframes. On the other hand, when the shorter the text were the more likely we found no or a single Spoken Timeframe. We also took this chance to evaluate manually the performance of our algorithm. In order to do that we started for each text identifying to sentences and how must they be grouped. We were able to determine that most Spoken Timeframe within the same Narrative Timeframe are usually completing the same idea or belong to the same speech. Nothing can be concluded though, since the data is not enough and even though we did mention large news articles, the longest is around 2 pages, we need to apply our approach on books and narratives to be more representative. We are pleased to get the same output, but we got one specific article that had a direct speech inside a different direct speech. The punctuation and the overall structure of the text was complex for us while manually extracting the timeframe and the algorithm had a two missing elements. This issue is not frequent so it will be handled later on. For the repartition of the narrative, almost 43% had only a single timeframe and the rest had varying amount the maximum being 6 Narrative timeframe per text.

It is important to mention that some dissimilarities were noticed when comparing the manually extracted narrative timeframe and our algorithm’s output. To be more specific, we have two patterns that provoke the creation of a new Narrative Timeframe. The first one occurs when a temporal expression is identified in the beginning of a sentence. When that’s the case no errors were identified. On the other hand, the other pattern is more complex. We must have a previously detected pattern where the tenses were different from the previous one. The purpose is to identify an end of a flashback or flash forward. While the use of a time expression with no change in the tense is considered a time skip or time jump in which we don’t go back initial timeframe that occurred before the flashback or flash forward. The latter pattern generated new Narrative Timeframe when there was no need to generate a new timeframe. To be more specific, out of the 68 news articles that showed multiple timeframe 13 showed this pattern. Out of the 13 papers, only 5 showed an unnecessary Narrative Timeframe. This issue is not considered relevant for now since this will be solved in the timeframe temporal relation extraction in our future work. We will than relate the new timeframe to the previous timeframe or the one prior to it.

In order to evaluate the usefulness of our approach from an event ordering point of view, we decided to apply one of the related work event ordering algorithm to compare the performance with and without the timeframe approach. We selected the CAEVO event ordering [26] which is a CAscading EVent Ordering architecture, that using inferences to generate more temporal relationship. The cascade name reflects the effect inferences have on the relations. The model is built on the TimeBank Dense dataset and have the “Vague” relation along with “Before”, “After”, “Includes”, “Is Included”, and “Simultaneous”. Just to put into perspective the importance of Event Ordering and Timeframe Event Ordering, we made a representation that highlight their use.

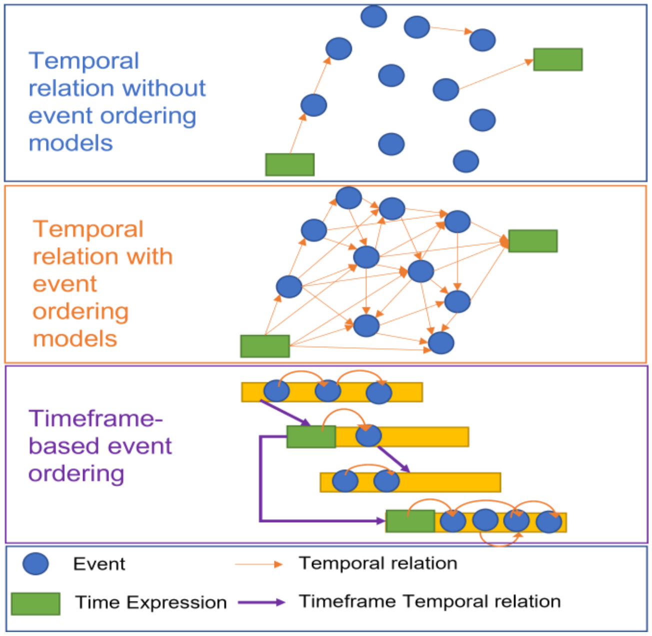


Figure 11. Temporal relations representation of multiple approaches.

Figure 11 shows the difference in the representation of temporal relations extract using multiple approaches. The first output shows a representation of an output without using an event ordering model. We can notice that hardly 4 events are connected and only 2 events are related to a time expression. The second presents the output of the same event extraction but after applying an event ordering model. This time most events are connected but the interpretation and the usability are complex. Finally, the third provides our desired representation using the temporal timeframes. This approach grouped events that occurred in the same period of time and limited the relation extraction between events from different timeframe. In this article we proposed the different type of timeframes and how to extract them without extracting the relationships between the timeframes. So in the current state of our approach, the Timeframe-based Event ordering is missing the temporal relation between the timeframes but using CAEVO, we evaluate the performance of the event ordering model intra-timeframe compared to not using our approach.

To elaborate on the event ordering evaluation, the first element done was evaluate the number of event which is identical in both, since the timeframe approach keep all the sentences in the text but segments them onto multiple timeframes. The temporal relation extracted highlighted the importance of our approach. Without the timeframe-based approach, on average more than 70% of the temporal relations were labeled “Vague” per text. The other most frequent relations were “After” and “Before”. The rest of the temporal relations were really rare, 36% of the text only had the dominant relations, and the percentage of the three remaining relations were less than 4% in the text. Having an average of 73 temporal relations per document, after using timeframes, the average number of temporal relations dropped to 29. The most frequent temporal relations were the “Before” and “After” relations 61% of the text.

We had notice 7 articles where no change was detected, this is due to the fact that these texts had no Spoken Timeframes and a single Narrative Timeframe. The rest of the articles that had a single Narrative Timeframe showed a difference in the results due to the presence of Spoken Timeframe. When multiple narrative timeframes were identified we also noticed I a small decrease in the “non-vague” relations. This decrease is due to the fact that the relation between the time expression that led to the Narrative Timeframe split and the previous sentence were broken. These temporal relations are not considered lost since they will be restored once the timeframe to timeframe relations are added to the approach. But the best element is the “Vague” relations between two timeframes are no longer present. In the following section, we will present possible ways to generate the relation between the timeframes.

VI. Future Work

After distributing the text onto the different timeframe the main target becomes in identifying the temporal relationship between the different timeframes available. Our approach adds a new type of temporal relation which is the relation between two timeframes. It is important to note that we identified three main type of timeframe to timeframe relations:

- Publication timeframe to Narrative Timeframe
- Spoken Timeframe to Narrative Timeframe
- Narrative Timeframe to Narrative Timeframe

In this perspective, the publication timeframe will have a direct relation with the first narrative timeframe in the text. The tense of the verbs, along with the time expression when mentioned at the beginning of the narrative timeframe will be used to identify if the narrative timeframe is about event that happened ‘Before’, ‘After’, or ‘During’ the Publication Timeframe. The same goes for the Spoken Timeframe and the Narrative Timeframe. The Spoken Timeframe will only be related to the Narrative Timeframe it occurred in. The tense and temporal expression used will enable the value assigned to the relation. The last relation which aims to join two Narrative Timeframes is more complexed. We can have relations between consecutive Narrative timeframe and none consecutive. In the current state of our Narrative Timeframe generation, two triggers can generate the creation of a Narrative Timeframe. The first one is the presence of a time expression at the beginning of a sentence. The time expression will be used in this case for identifying the relation with the previous timeframe. When a specific date is mention a relation with the Publication timeframe can be added. The most complexed part is when a new timeframe is generated due to verb tense changes, in the specific pattern. For now, with no time expression and only verb tense to compare, the relation will be associated to the previous Narrative Timeframe and the first timeframe that has common verb tenses. For now, the relation generated will be labeled “Vague” since the tenses are not enough to deduct the relations. The proposed approach and criteria to extract

Timeframe relation is more developed and presented in Matta et al. [43].

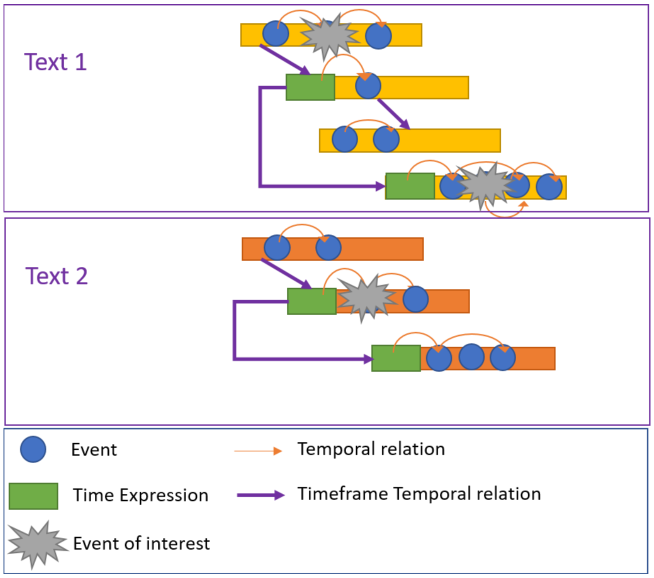


Figure 12. Intra-text and extra-text event desambiguation.

The last step in our work will be event disambiguation. We can notice in Figure 12, the event of interest that is mentioned in two different text, and mentioned twice in the same text. We intent to use Event Coreference Resolution, to identify the mention of the same event in the same text and use the timeframes to identify the change in state and entities of these events through text. And later on, identify the mention of specific event in multiple text, and using publication timeframes, narrative timeframes, and the relations between them, identify how the event of interest is progressing, or even analyses how object, entities are evolving through time. In that case, we would consider a person, an object, an organization, and view the events that are related to them.

VII. Conclusion

Finally, event extraction is an essential task in the Natural Language Processing field. It enables the use of text data in order to build decision-making systems and for event monitoring. It also enables story follow up, first story detection, event extraction along with the entities participating in the events. Those event can also be represented onto graphs, event centric or entity centric in order to have a clear visualization of the identified information. Event ordering is a branch in Event Extraction that focus on the identification of the temporal relation between the different event extracted. In this paper, we highlighted the need for timeframes to improve event ordering in the event extraction field. Three types of timeframes were presented: The Publication, the Narrative, and the Spoken Timeframe. Publication Timeframes will be used for multiple text analysis as a temporal indicator of the text. Narrative Timeframes enable the distinguishing of multiple periods of time used in a text, notably when a flashback or a flash-forward occurs. Finally, the Spoken Timeframe enables the distinction between the Narrative Timeframes and the timeframe of "spoken" elements in a text, such as announcements or dialogs. We set a few semantic patterns for the identification and extraction of the different timeframes. In future work section, we provided possible relation extraction methods for the timeframes and the events of each timeframe. Some of the process and criteria had been introduced but are still in their early stages. We intend to evaluate the performance on longer texts and a larger number. We will also be distinguishing the multiple classes of events: point, process, culmination, and culminated process in order to identify states available in timeframes. This work will complete our study on detection and representation context from text [44].

References

[1] N. Matta, N. Matta, N. Declercq, and A. Marcante, 'Semantic Patterns to Structure TimeFrames in Text', INTELLI 2022, May 2022, pp. 16–23. Accessed: Sep. 13, 2022. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=intelli_2022_1_40_60016

[2] X. Wu, T. Wang, Y. Fan, and F. Yu, 'Chinese Event Extraction via Graph Attention Network', ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 21, no. 4, p. 71:1-71:12, Jan. 2022, doi: 10.1145/3494533.

[3] W. Zhao, J. Zhang, J. Yang, T. He, H. Ma, and Z. Li, 'A novel joint biomedical event extraction framework via two-level modeling of documents', Inf. Sci., vol. 550, pp. 27–40, Mar. 2021, doi: 10.1016/j.ins.2020.10.047.

[4] J. Zagal and M. Mateas, 'Time in Video Games: A Survey and Analysis', Simul. Gaming - Simulat Gaming, vol. 41, Jan. 2010, doi: 10.1177/1046878110375594.

[5] F. Hogenboom, F. Frasincar, U. Kaymak, F. de Jong, and E. Caron, 'A Survey of event extraction methods from text for decision support systems', Decis. Support Syst., vol. 85, pp. 12–22, May 2016, doi: 10.1016/j.dss.2016.02.006.

[6] Q. Li et al., 'Reinforcement Learning-Based Dialogue Guided Event Extraction to Exploit Argument Relations', IEEEACM Trans. Audio Speech Lang. Process., vol. 30, pp. 520–533, 2022, doi: 10.1109/TASLP.2021.3138670.

[7] W. Xiang and B. Wang, 'A Survey of Event Extraction From Text', IEEE Access, vol. 7, pp. 173111–173137, 2019, doi: 10.1109/ACCESS.2019.2956831.

[8] M. J. Buehner and J. May, 'Knowledge mediates the timeframe of covariation assessment in human causal induction', Think. Reason., vol. 8, no. 4, pp. 269–295, Nov. 2002, doi: 10.1080/13546780244000060.

[9] W. H. Newton-Smith, 'The Structure of Time', Routledge & CRC Press, Jun. 01, 2020. https://www.routledge.com/The-Structure-of-Time/Newton-Smith/p/book/9781138394063 (accessed Feb. 09, 2022).

[10] M. Moens and M. Steedman, 'Temporal Ontology and Temporal Reference', Comput. Linguist., vol. 14, no. 2, pp. 15–28, 1988.

[11] Z. Vendler, 'Verbs and Times', Philos. Rev., vol. 66, no. 2, pp. 143–160, 1957, doi: 10.2307/2182371.

[12] Z. Vendler, Linguistics in Philosophy. Cornell University Press, 2019. doi: 10.7591/9781501743726.

[13] X. Duan et al., 'Temporality-enhanced knowledgememory network for factoid question answering', Front. Inf. Technol. Electron. Eng., vol. 19, no. 1, pp. 104–115, Jan. 2018, doi: 10.1631/FITEE.1700788.

[14] Y. Chen, Z. Ding, Q. Zheng, Y. Qin, R. Huang, and N. Shah, 'A History and Theory of Textual Event Detection and Recognition', IEEE Access, vol. 8, pp. 201371–201392, 2020, doi: 10.1109/ACCESS.2020.3034907.

[15] M. Mejri and J. Akaichi, 'A Survey of Textual Event Extraction from Social Networks', 2017.

[16] H. Zhou, H. Yin, H. Zheng, and Y. Li, 'A survey on multi-modal social event detection', Knowl.-Based Syst., vol. 195, p. 105695, May 2020, doi: 10.1016/j.knosys.2020.105695.

[17] G. Jacobs and V. Hoste, 'SENTiVENT: enabling supervised information extraction of company-specific events in economic and financial news', Lang. Resour. Eval., vol. 56, no. 1, pp. 225–257, Mar. 2022, doi: 10.1007/s10579-021-09562-4.

[18] J. Liu, L. Min, and X. Huang, 'An overview of event extraction and its applications', ArXiv211103212 Cs, Nov. 2021, Accessed: Jul. 18, 2022. [Online]. Available: http://arxiv.org/abs/2111.03212

[19] S. Li, H. Ji, and J. Han, 'Document-Level Event Argument Extraction by Conditional Generation', ArXiv210405919 Cs, Apr. 2021, Accessed: Aug. 05, 2021. [Online]. Available: http://arxiv.org/abs/2104.05919

[20] F. Li et al., 'Time event ontology (TEO): to support semantic representation and reasoning of complex temporal relations of clinical events', J. Am. Med. Inform. Assoc., vol. 27, no. 7, pp. 1046–1056, Jul. 2020, doi: 10.1093/jamia/ocaa058.

[21] Q. Ning, Z. Feng, and D. Roth, 'A Structured Learning Approach to Temporal Relation Extraction', ArXiv190604943 Cs, Jun. 2019, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1906.04943

[22] C.-G. Lim, Y.-S. Jeong, and H.-J. Choi, 'Survey of Temporal Information Extraction', J. Inf. Process. Syst., vol. 15, no. 4, pp. 931–956, 2019, doi: 10.3745/JIPS.04.0129.

[23] A. Leeuwenberg and M.-F. Moens, 'Structured Learning for Temporal Relation Extraction from Clinical Records', in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain, Apr. 2017, pp. 1150–1158. Accessed: Feb. 09, 2022. [Online]. Available: https://aclanthology.org/E17-1108

[24] Q. Ning, Z. Feng, H. Wu, and D. Roth, 'Joint Reasoning for Temporal and Causal Relations', ArXiv190604941 Cs, Jun. 2019, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1906.04941

[25] J. Pustejovsky et al., 'The TimeBank corpus', Proc. Corpus Linguist., Jan. 2003.

[26] N. Chambers, T. Cassidy, B. McDowell, and S. Bethard, 'Dense Event Ordering with a Multi-Pass Architecture', Trans. Assoc. Comput. Linguist., vol. 2, pp. 273–284, Oct. 2014, doi: 10.1162/tacl_a_00182.

[27] S. Bethard, 'ClearTK-TimeML: A minimalist approach to TempEval 2013', in Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, USA, Jun. 2013, pp. 10–14. Accessed: Feb. 07, 2022. [Online]. Available: https://aclanthology.org/S13-2002

[28] N. Chambers, 'NavyTime: Event and Time Ordering from Raw Text', in Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, USA, Jun. 2013, pp. 73–77. Accessed: Jun. 07, 2022. [Online]. Available: https://aclanthology.org/S13-2012

[29] H. Llorens, L. Derczynski, R. Gaizauskas, and E. Saquete, 'TIMEN: An Open Temporal Expression Normalisation Resource', in Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, May 2012, pp. 3044–3051. Accessed: Apr. 01, 2022. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/128_Paper.pdf

[30] N. Laokulrat, M. Miwa, Y. Tsuruoka, and T. Chikayama, 'UTTime: Temporal Relation Classification using Deep Syntactic Features', in Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, USA, Jun. 2013, pp. 88–92. Accessed: Aug. 07, 2022. [Online]. Available: https://aclanthology.org/S13-2015

[31] P. K. Choubey and R. Huang, 'A Sequential Model for Classifying Temporal Relations between Intra-Sentence Events', ArXiv170707343 Cs, Jul. 2017, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1707.07343

[32] P. Mirza and S. Tonelli, 'CATENA: CAusal and TEmporal relation extraction from NAtural language texts', p. 12, 2016.

[33] T. Caselli and P. Vossen, 'The Event StoryLine Corpus: A New Benchmark for Causal and Temporal Relation Extraction', in Proceedings of the Events and Stories in the News Workshop, Vancouver, Canada, Aug. 2017, pp. 77–86. doi: 10.18653/v1/W17-2711.

[34] P. Chen, K. Liu, Y. Chen, T. Wang, and J. Zhao, 'Probing into the Root: A Dataset for Reason Extraction of Structural Events from Financial Documents', in Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, Apr. 2021, pp. 2042–2048. doi: 10.18653/v1/2021.eacl-main.175.

[35] P. Mathur, R. Jain, F. Dernoncourt, V. Morariu, Q. H. Tran, and D. Manocha, 'TIMERS: Document-level Temporal Relation Extraction', in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Online, Aug. 2021, pp. 524–533. doi: 10.18653/v1/2021.acl-short.67.

[36] A. Pal, J. Margatan, and J. Konstan, 'Question temporality: identification and uses', in Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, New York, NY, USA, Feb. 2012, pp. 257–260. doi: 10.1145/2145204.2145246.

[37] T. Kwiatkowski et al., 'Natural Questions: A Benchmark for Question Answering Research', Trans. Assoc. Comput. Linguist., vol. 7, pp. 453–466, Aug. 2019, doi: 10.1162/tacl_a_00276.

[38] J. Hamann and L. Suckert, 'Temporality in Discourse: Methodological Challenges and a Suggestion for a Quantified Qualitative Approach', Forum Qual. Sozialforschung Forum Qual. Soc. Res., vol. Vol 19, p. No 2 (2018), Mar. 2018, doi: 10.17169/FQS-19.2.2954.

[39] V. Moutinho, João Cordeiro, and P. Brazdil, 'Association and Temporality between News and Tweets', pp. 500–507, Sep. 2019, doi: 10.5220/0008362105000507.

[40] T. Poell, 'Social media, temporality, and the legitimacy of protest', Soc. Mov. Stud., vol. 19, no. 5–6, pp. 609–624, Nov. 2020, doi: 10.1080/14742837.2019.1605287.

[41] S. W. Brown, J. Bonn, J. Gung, A. Zaenen, J. Pustejovsky, and M. Palmer, 'VerbNet Representations: Subevent Semantics for Transfer Verbs', in Proceedings of the First International Workshop on Designing Meaning Representations, Florence, Italy, Aug. 2019, pp. 154–163. doi: 10.18653/v1/W19-3318.

[42] E. Partalidou, E. Spyromitros-Xioufis, S. Doropoulos, S. Vologiannidis, and K. I. Diamantaras, 'Design and implementation of an open source Greek POS Tagger and Entity Recognizer using spaCy', in 2019 IEEE/WIC/ACM

International Conference on Web Intelligence (WI), Oct. 2019, pp. 337–341.

[43] N. Matta, N. Matta, N. Declercq, and A. Marcante, Evolution Discovery in Textual Data. 18th Annual International Conference on Information Technology & Computer Science (ATINER 2022); in press.

[44] N. Matta, N. Matta, E. Giret, and N. Declercq, ‘Enhancing Textual Knowledge Discovery using a Context-Awareness Approach’, in 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Dec. 2021, pp. 233–237. doi: 10.1109/CSCI54926.2021.00071.