text2story: A Python Toolkit to Extract and Visualize Story Components of Narrative Text

Evelin Amorim[1], **Ricardo Campos**[1,2], **Alípio Jorge**[1,3], **Pedro Mota**[3], **Rúben Almeida**[1]
[1]INESC TEC
Porto, Portugal
{evelin.f.amorim,ricardo.campos,amjorge,ruben.f.almeida}@inesctec.pt

[2]University of Beira Interior
Covilhã, Portugal

[3]University of Porto
Porto, Portugal
pedndm@gmail.com

Abstract

Story components, namely, events, time, participants, and their relations are present in narrative texts from different domains such as journalism, medicine, finance, and law. The automatic extraction of narrative elements encompasses several NLP tasks such as Named Entity Recognition, Semantic Role Labeling, Event Extraction, and Temporal Inference. The text2story Python, an easy-to-use modular library, supports the narrative extraction and visualization pipeline. The package contains an array of narrative extraction tools that can be used separately or in sequence. With this toolkit, end users can process free text in English or Portuguese and obtain formal representations, like standard annotation files or a formal logical representation. The toolkit also enables narrative visualization as Message Sequence Charts (MSC), Knowledge Graphs, and Bubble Diagrams, making it useful to visualize and transform human-annotated narratives. The package combines the use of off-the-shelf and custom tools and is easily patched (replacing existing components) and extended (e.g. with new visualizations). It includes an experimental module for narrative element effectiveness assessment and being is therefore also a valuable asset for researchers developing solutions for narrative extraction. To evaluate the baseline components, we present some results of the main annotators embedded in our package for datasets in English and Portuguese. We also compare the results with the extraction of narrative elements by GPT-3, a robust LLM model.

Keywords: narrative extraction, toolkit, framework, python, text annotation, information retrieval

1. Introduction

A narrative is usually understood by linguists as a sequence of events that are related to each other(Toolan, 2012). The concept of an event can encompass other attributes, like when it occurred (time) and who took part in it (participants). The way events and their attributes are arranged aids the comprehension of the main information of a text. Thus, such a structure is pervasive in different text genres and its automatic extraction can benefit several areas of application, such as journalism (Campos et al., 2021), finance (El-Haj et al., 2022) and health (Jindal and Roth, 2013).

An approach to automatically extract narratives can start with the identification of the events, participants, temporal aspects, and the relationships between them. First, human experts annotate the components of narrative text and then perform some analysis of data. Next, a machine learning model is designed and trained on the labeled text. This sequence of tasks generally requires combining different types of tools, which can be a cumbersome task. To smooth the process of automatically extracting narratives, we propose the text2story python toolkit with the following modules: (1) An annotator module that already comprises off-the-shelf tools as baselines to annotate automatically text, besides an infra-structure to implement customized annotators; (2) A reader module that provides classes to read some well-known annotation format files; (3) An experiments module that automatizes batch experiments and their evaluations; (4) A visualization module that currently produces three types of visual representation of annotation, namely, Message Sequence Chart (MSC), Knowledge Graphs (KG), and Bubble Diagrams (BD). The combination of these tools to extract the narrative components poses some challenges, for instance, managing dependencies in Python projects can become problematic, especially when dealing with multiple dependencies in a project (Wang et al., 2022, 2020). Also, differences in programming interfaces can complicate code comprehension, hindering research progress in the field. Our tool simplifies this process, combining different off-the-shelf tools, and

LREC-COLING 2024, pages 15761–15772

providing a simple and accessible programming interface. The user is also allowed to extend annotation modules, and reading data.

The text2story has three main cornerstones to support its main goal, to assist with the narrative extraction task. The first cornerstone is regarding how each tool extracts the narrative structure. The second cornerstone is the visual representation of labels. The third cornerstone is the batch experiments. Considering the first cornerstone, some programming libraries have been proposed over the years to extract specific narrative components such as events or time expressions from the text while ignoring the narrative structure as a whole. For instance, Zhang et al. (2022) presents a Python toolkit called DeepKE to extract named entities, events, and relations between entities from a text and then populate a knowledge base. Another tool based on a web API is presented by Wen et al. (2021), which builds a temporal event graph from a collection of documents. A more comprehensive toolkit introduced by Jin et al. (2021) aims to extract both entities and their relationships. The second cornerstone, the visual representation of annotations, can be covered by some annotation tools like the general-purpose annotation tool BRAT (Stenetorp et al., 2012), and others like Prodigy (Montani and Honnibal, 2018) that includes automatic labeling of some narrative components, like Named Entities and events. In the context of the narrative structure, CATMA'S (Bögel et al., 2015; Horstmann, 2020) is a web application whose goal is to label literary texts. Finally, the third cornerstone, the batch experiments, is partially covered by annotation tools like Prodigy which has Inter Annotator Metrics, and DeepKE which has some traditional metrics for Information Extraction tasks. However, none of these tools integrate all three cornerstones into a single programming toolkit. We hope that these contributions help researchers advance the results of the narrative extraction task.

As a contribution to the community, we are publicly releasing our library's code as a pip python package[1]. In addition to that, we publish a video demonstration[2] and one Python notebook that presents the main features of the toolkit[3]. In the next sections, we will detail the architecture and the pipeline for narrative extraction, the process to produce the visual representations with the text2story toolkit, and the results achieved by each baseline module in English and Portuguese datasets. We conclude in Section 5 with some final remarks and future work.

[1] https://pypi.org/project/text2story/

[2] https://youtu.be/VUcxKhYA3lc

[3] https://bit.ly/3Fq9JK1

2. The text2story Architecture

The main modules of the toolkit are depicted in Figure 1. In the following, we provide some additional details about each one of them.

- **Core.** The main class in this module is the Narrative class, which is composed of the entities' objects and the relations between them. The class Entity Structures comprises the Participants, Events, and Time expression types. The Link Structures class defines the links between all the main elements of a narrative. The Annotator class defines a pipeline to automatically label the components of a narrative text. Finally, there are exceptions related to the labeling of text that the system can raise.

- **Annotators.** In this module, there are off-the-shelf annotators, like AllenNLP(Gardner et al., 2018), Heildeltime(Strötgen and Gertz, 2010)[4], NLTK(Loper and Bird, 2002), Spacy(Honnibal et al., 2020), tei2go (Sousa et al., 2023a), and a BERT model for recognition of Named Entities in the Portuguese language[5]. Regarding the AllenNLP module, there are different models available, thus we employed the model developed by Oliveira et al. (2021)[6] for the Portuguese language, which is based on transformers, and for the English we also use a BERT-based model, which was developed by Shi and Lin (2019)[7]. Users can also define their own annotators in this module.

- **Readers.** The reading of data is performed by one of the classes of this module. If the annotations were manually performed by BRAT, like in Figure 3a, then a class dedicated to this kind of format can process them. In addition to that, there are already classes devoted to reading some common format corpus, like ECB+ (Cybulska and Vossen, 2014), Propbank (Kingsbury and Palmer, 2002), FrameNet (Baker et al., 1998),

[4] In particular, we used the Python Wrapper of Heildeltime, available at https://github.com/JMendes1995/py_heideltime.

[5] We specifically employed the following model https://huggingface.co/arubenruben/NER-PT-BERT-CRF-Conll2003

[6] This and other models for the Portuguese language are available at https://github.com/asofiaoliveira/srl_bert_pt

[7] The version of the model we use is available at https://storage.googleapis.com/allennlp-public-models/structured-prediction-srl-bert.2020.12.15.tar.gz

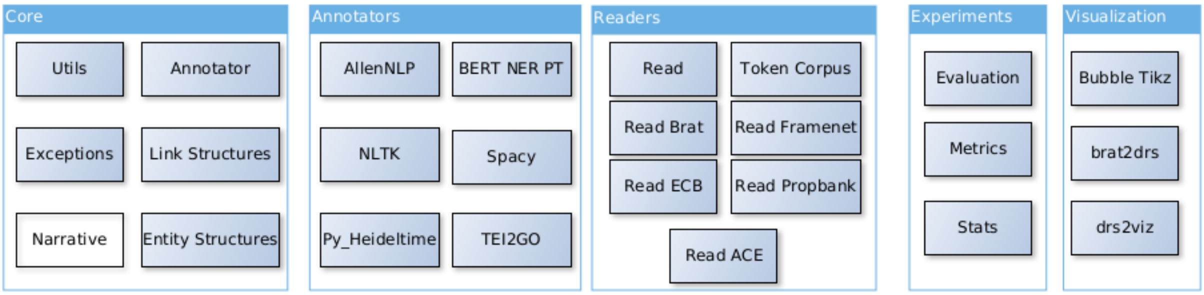


Figure 1: Main modules of text2story toolkit.

and ACE (Doddington et al., 2004). In this module, users can define their own readers by following the guidelines of the abstract class Read.

- **Experiments.** To aid possible benchmarks, we add the experiments module, which has the classes Metrics with the most common classification metrics; Evaluation class that facilitates assembling batch experiments, and Stats class that has methods to analyze the dataset, i.e., its main statistics.

- **Visualization.** The visualization module is responsible for producing a visual representation of annotations, whether manually or automatically. There are three types of visualizations in tex2story: Message Sequence Chart (MSC), Knowledge Graph (KG), and Bubble Diagrams (BD). The first two employ an intermediate logical language called Discourse Representation Structure (DRS) (Geurts et al., 2020) to build unambiguous representations, and also to infer relations between entities. After the conversion of the annotation file to the DRS file format, which is produced by the submodule `brat2drs`, these two types of visualizations can be built by the submodule `drs2viz`. The Bubble Diagram is produced by the submodule `bubble_tikz`. We plan to work on integrating a conversion for DRS in this visualization as future work.

3. The Narrative Extraction and Visualization

The main purpose of the text2story toolkit is to perform extraction and visualization of the main components of a narrative. Therefore, in the next subsections, we explain (1) how to assemble a pipeline for narrative extraction, and (2) how visualization is produced from a pipeline result.

3.1. A Pipeline for the Narrative Extraction

The pipeline workflow for narrative extraction using the text2story toolkit is illustrated in Figure 2. The first step is to input a raw text to extract the main entities of a narrative, i.e., participants, events, and time. Next, the extraction of semantic links between participants and events is performed. Then, an annotated file can be saved. Consider the following Example 1 as the value of `doc` variable in the code.

Example 1. *Mrs Potter was Mrs Dursley's sister, but they hadn't met for several years; in fact, Mrs Dursley pretended she didn't have a sister, because her sister and her good-for-nothing husband were as unDursleyish as it was possible to be.*

```python
# these are some imports
import text2story as t2s

narrative_doc = t2s.Narrative("en",doc,"2023") # this is the narrative object
participants = narrative_doc.extract_participants("spacy")
times = narrative_doc.extract_times("py_heideltime")
events = narrative_doc.extract_events("allennlp")
semanticrole_links = narrative_doc.extract_semantic_role_links()
```

Code Python for the Extraction of Narrative of Example 1

Observe that the interface of our programming library to extract the narrative from the text example is simple and is done with roughly six lines of code. A more comprehensive tutorial can be seen in our Colab notebook [8] which also demonstrates how to set the environment for the proposed toolkit. Figure 3b illustrates an example of an annotated excerpt from Example 1. In Figure 3b, it is possible to observe that the automatic annotator fails

[8] https://bit.ly/3Fq9JK1

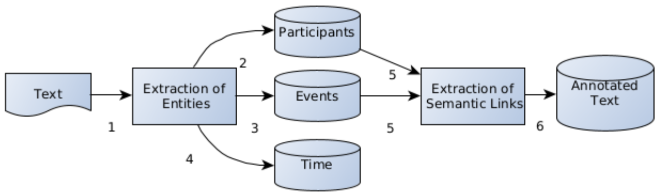


Figure 2: The main steps of a pipeline for the Narrative Extraction in the text2story toolkit

to identify the participant "they" and captures the complete participant "Mrs Dursley's sister". Also, it incorrectly identified semantic links and missed one temporal link. The remaining entities were correctly identified. For the full manual annotations of this example, we refer the reader to Figure 7 in Section A.

3.2. The Visualization of Annotations

After performing annotation using one of the built-in annotators of the text2story toolkit or a human annotator to label some data, the proposed tool can construct three types of visualizations, namely, a Message Sequence Chart (MSC), a Knowledge Graph (KG) and a Bubble Diagram (BD). The input of our visualization pipeline is an annotated file in BRAT format. The first step in the pipeline is, based on the annotation, to build an output in the Discourse Representation Structure (DRS) language (Kamp and Reyle, 1993; Bos et al., 2017). This step is relevant since DRS is a logical language that helps to provide an unambiguous representation of the narrative elements and their relations. Additionally, it is possible to reason on top of these elements, which allows us to capture further elements or relations. Finally, there is the processing of the DRS file, which produces a visual representation of the annotation as a Message Sequence Chart (MSC) or as a Knowledge Graph (KG).

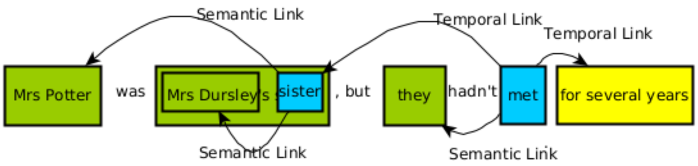
These types of diagrams represent participants and their relationships with other participants. For instance, consider the sentence "Mrs Potter was Mrs Dursley's sister, but they hadn't met for several years; in fact, Mrs Dursley pretended she didn't have a sister, because her sister and her good-for-nothing husband were as unDursleyish as it was possible to be.", which is part of the first Harry Potter book series. We applied in this sentence the pipeline steps depicted in Figure 2. To automatically annotate participants, we employed the Spacy NER[10], to annotate time expression the Heideltime was used, and to annotate events and semantic relations, we employed the Structured predictor English Model of the AllenNLP[11]. The outcome is an annotated file that is converted to a DRS file, which is, then used to build the MSC and KG representation. The final outputs of the given example are depicted as MSC in Figure 4 and as KG in Figure 5.

In the MSC visualization, lifelines representing entities are the identified participants. However, the chart depicts only one mention per participant, otherwise redundant information could be represented in the visualization. In Figure 4, the pronoun "her" represents four extracted participants. Additionally, in the same example, the event annotator recognizes "was" as an event and the semantic link annotator links it with two participants. Consequently, these two participants are connected through this event, possibly assuming different roles. This connection is evident in the MSC, where "Mrs. Potter" and "her" share the same link, signifying that "her" encompasses "Mrs. Dursley's". The representations of the other components, lifelines, and interactions among them, follow the same logic. Compared to the manual annotations (Figure A), it is possible to observe that the automatic annotation detected few participants. For instance, "her sister and her good-for-nothing husband" is considered as four different participants by the human annotator, "her", "sister", "her", and "good-for-nothing husband". Thus, while in the automatic annotation, there are seven participants, in the manual annotation there are eleven participants. Some events were wrong as well. The human annotator does not con-

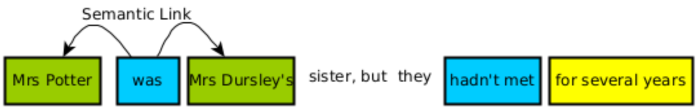
[9]Note that the auxiliary verb and the negation "hadn't" are not annotated since the guidelines of annotation expressly state that auxiliary verbs should not be noted as events. However, the negation is considered as an attribute of the event "met", which indicates the Polarity of the event. This attribute indicates if an event is a "Negative" event or a "Positive" event. In this example, "met" has the polarity attributed annotated as "Negative". For more information about the annotation scheme, we refer the reader to the paper (Silvano et al., 2021).

[10]In our tests, we use the model available in `https://github.com/explosion/spacy-models/releases/tag/en_core_web_lg-3.2.0`

[11]Specifically, we employ the model available in `https://storage.googleapis.com/allennlp-public-models/structured-prediction-srl-bert.2020.12.15.tar.gz`



(a) Human labeling snippet text[9]



(b) Automatic labeling snippet text

Figure 3: Human and Automatic labeling text for an excerpt of The First Book of Harry Potter Series (The original sentence is in Example 1, but for better visual readability of the annotations we presented only the first clause of the sentence, for the full sentence annotation see Figure 7).

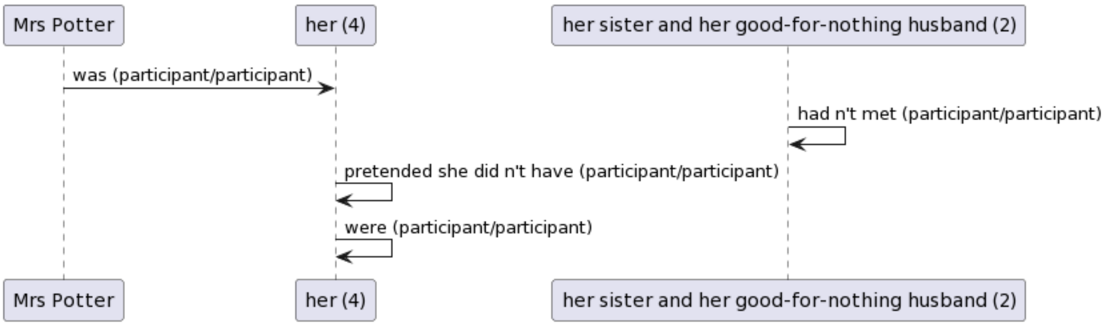


Figure 4: MSC representation built from the automatic labeling of a sentence of Harry Potter's book. For the manual annotation, see Figure 7.

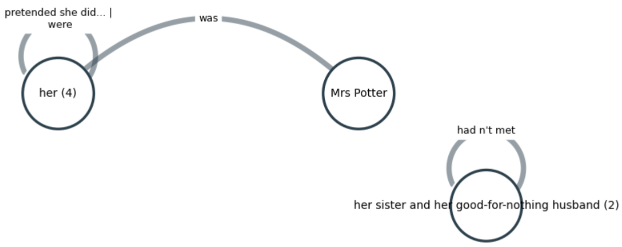


Figure 5: The Knowledge Graph Representation built from the automatic labeling of a sentence of Harry Potter's book. For the manual annotation, see Figure 7.

sider "were" and "was" as events, while the automatic annotator considers them. The events "pretended" and "met" were also annotated along with other tokens that the human annotator did not take into account.

The graph visualization employs a similar logic to MSC but does not take into account the order of participant appearance. The idea is to give an overview of who the participants are and with whom they relate. In Figure 5, there are three participants, of whom only two are related to each other. The mistakes of the automatic annotators were the same as in the MSC figure since we employed the same engine and pipeline. Despite the several errors of the automatic annotator, the reader should keep in mind that we employed a Semantic Labeling Role (SRL) model for a different task (narrative extraction). Hence, this model can be a baseline for this task, and for future work, we intend to embed models specific to the narrative extraction task.

Unlike MSC and KG, the Bubble Diagram's primary purpose is to represent connected events of type "Reporting", which divides the narrative into two layers. In this diagram, one layer of the narrative is represented by the Big Bubble, and the little bubbles represent another layer (for more information about the layer of reporting events see (Silvano et al., 2023)). This kind of scheme also repre-

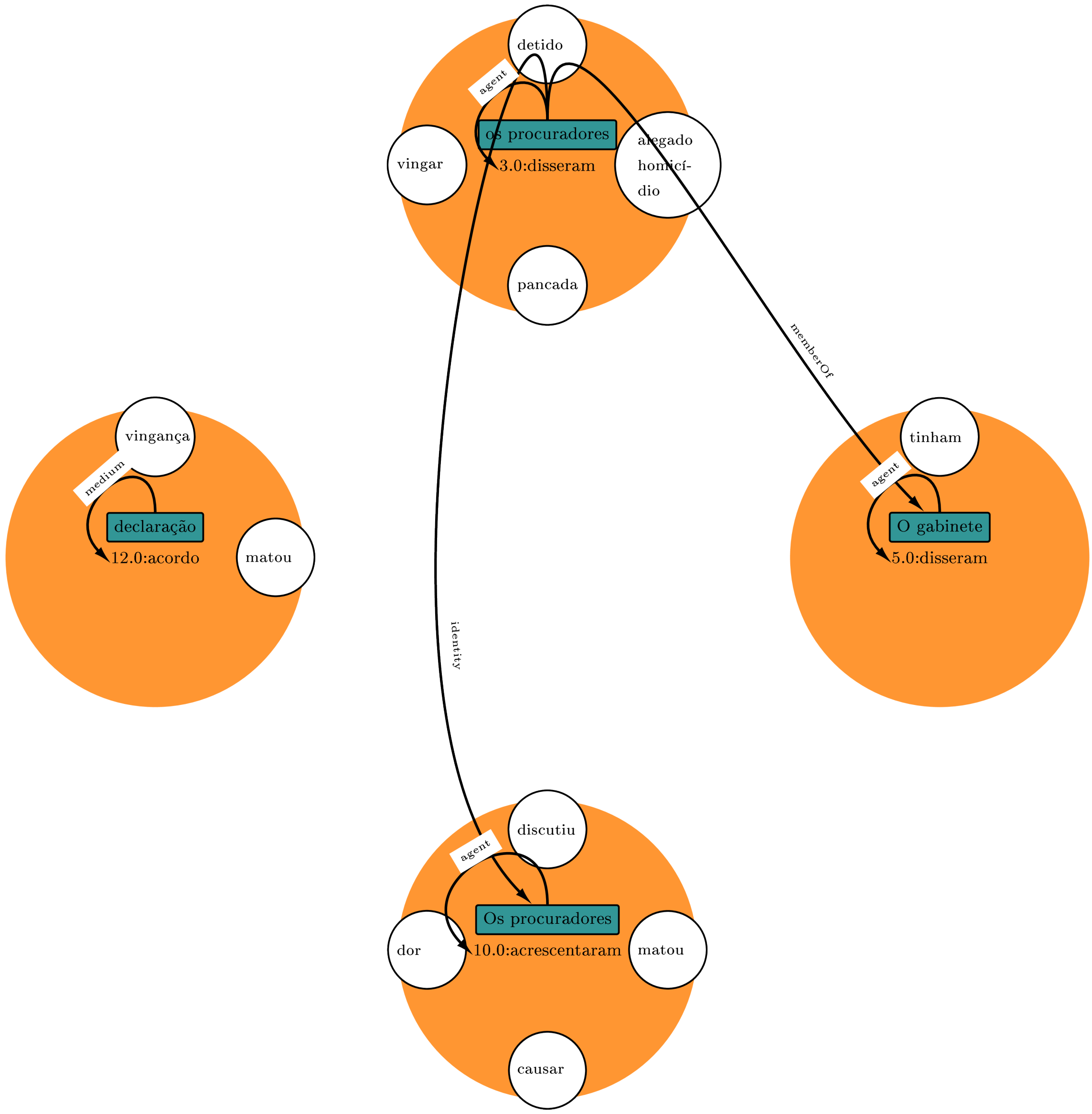


Figure 6: An example of a Big Bubble representation for the sentence *"Aparentemente, numa vingança contra a mulher, matou os filhos", de acordo com uma declaração publicada pelo gabinete.* (Apparently, in revenge against his wife, he killed the children, according to a statement published by the office.). The bubbles follow a chronological order. The first big bubble, representing a reporting event, is positioned at 12 o'clock, and the subsequent big bubbles, following the hourly pattern, represent reporting events that occur later. This allows us to discern the sequence of reporting events in the text based on the order of the big bubbles. Each reporting event also contains events within it. These events are the ones that have been declared or reported by someone and are also arranged chronologically, similar to the big bubbles. The agent reporting the events is also depicted in the figure by a green rectangle at the center of the large bubble. Finally, the semantic relationships between these events and the participants are depicted through arrows connecting the bubbles or rectangles. In this example, the reporting event is *acordo* (according to), whose medium (a type of participant) is *declaração* (statement) and includes two other events: *vingança* (revenge) and *matou* (killed)

sents the temporal links between the events. One use of this kind of diagram is to analyze events of type "Reporting", which is a common class of events in news text. Silvano et al. (2023) employed this visual representation to analyze reporting events in a set of Portuguese news data. In Figure 6, there is one example of what is a Big Bubble (representing the event *acordo*) and the Little Bubbles that it includes, the events *vingança* (revenge) and *matou* (killed). Since this kind of visualization requires a specified class of event, it is necessary to manually label the class of the event or employ a customized classifier to this end. The example of Figure 6 was annotated by a human, and it is only an excerpt from a news dataset that was analyzed by Silvano et al. (2023). Such kind of representation can contain more Big Bubbles, however, due to the limited space we present only a part of the visual representation. To see the full figure, we refer the reader to our Colab notebook which produces a BD using the text2story toolkit[12]. Also, other types of events can be a Big Bubble, in which case only the specified type has to change.

4. Experiments

We tested the baselines of our pipeline in two different datasets, in the ACE 2005 dataset (Doddington et al., 2004) in the English Language and the Lusa News dataset (Silvano et al., 2021) in the Portuguese language. Additionally, we compared the baseline of each extraction component in the text2story pipeline with a Large Language Model (LLM) to provide context for the results related to these robust models, which, however, do not yet have an infrastructure like our pipeline to facilitate narrative element extraction. In the following subsections, we detail the datasets and results achieved.

4.1. Datasets

The ACE 2005 is a well-known English dataset for the extraction of events. Besides events, in ACE, the annotations also comprise entities that participate in the labeled events, time expressions, and some relations between all those elements. According to the ISO Semantic Annotation Framework (ISO-24617-9, 2019), the ACE annotation only considers links of type objectal links. These objectal links establish connections between entities that are related based on extra-linguistic concepts. This implies that these entities are linked in the real-world context, regardless of the specific language used within the text.

The Lusa News is a Portuguese dataset of manually labeled news. The annotation procedure follows the guidelines described by Silvano et al.

[12]https://bit.ly/3Fq9JK1

Narrative Component	ACE		Lusa News	
	Train	Test	Train	Test
Participants	5,948	37,071	622	2,644
Events	585	3,692	524	2,332
Times	670	3,700	67	338
#token	34,208	213,273	3,707	16,805
#documents	80	455	20	90

Table 1: Datasets statistics

(2021), which is based on ISO standards (ISO-24617-9, 2019). The main elements annotated in the Lusa News are events, participants, time expressions, semantic roles links, and objectal links. Table 1 describes the amount of each one of the main narrative elements annotated in these two datasets. Each dataset consists of two parts: a "training" split and a "testing" split. The first split was employed in developing the prompt for the LLMs tested, and the second split was used to test the text2story components and the LLMs[13].

4.2. Results

In this section, we describe the results of the extraction of events, participants, and time from text using the proposed package, and compare them with the results of the GPT-3 model. Next, we detail the test's experimental design and the metrics employed. Finally, we present tables with our results.

Experimental design. As mentioned in the previous section, we divided our dataset into the train and the test parts. For the tests using the text2story package, it is unnecessary to employ a training set since the baselines of its components only apply pre-trained models. Thus, we consider our baselines as zero-shot annotators.

The GPT-3 model, in contrast, requires a subset for prompt development. The methodology for the prompt construction is described by Sousa et al. (2023b). We also used the codebase[14] to extract narrative components from text using the LLMs. In the prompt experiments, we explored various configurations, and here we present the results for the best one. However, the codebase we used for the experiments with GPT-3 did not include functionality for extracting the links between narrative elements. Therefore, while our primary focus was on the extraction of all narrative elements and their relationships, the absence of event-participant links was a technical limitation rather than a deliberate omission. We acknowledge that this represents a potential avenue for future research, and we en-

[13]The availability of the splits is in the following links https://anonymfile.com/jz9l/slipt-ace.zip and https://anonymfile.com/QOe6/slipt-lusa.zip.

[14]`https://github.com/hmosousa/gpt_struct_me`

		ACE			Lusa News		
		P_r	R_r	F_{1_r}	P_r	R_r	F_{1_r}
Time	TEI2GO	**0.75**	**0.60**	**0.64**	0.70	**0.81**	**0.73**
	Heideltime	0.68	0.53	0.57	0.70	0.80	**0.73**
	GPT-3	0.61	0.44	0.46	**0.82**	0.52	0.61
Participants	SRL	0.29	0.02	0.08	**0.93**	0.15	0.26
	SPACY	**0.76**	0.25	0.36	0.77	0.33	0.45
	GPT-3	0.68	**0.52**	**0.56**	0.70	**0.77**	**0.72**
Events	SRL	0.10	**0.45**	**0.15**	**0.65**	0.37	**0.68**
	GPT-3	**0.16**	0.079	0.08	0.51	**0.71**	0.57

Table 2: Results for the Annotators of text2story modules and GPT-3 in the ACE 2005 and Lusa News datasets

courage further investigations into this aspect to provide a more comprehensive understanding of the potential of such LLMs in extracting links between narrative elements.

For the ACE dataset, we employed the `adj` annotations as it includes the validation by a third annotator in addition to the annotations by the other two human annotators. We also consider only the event trigger in the task of event detection. The event in the ACE dataset can comprise a whole sentence that can present the event arguments, as participants, and location, among others. Since the other two datasets label only the event triggers as events, we decided to identify only this element in ACE dataset.

Metrics. To evaluate our results, we use the metrics *Precision*, *Recall*, and *f1*. However, we apply these metrics in two different ways. As proposed by UzZaman et al. (UzZaman et al., 2013), there are two versions of these metrics, the strict and the relaxed. We apply the relaxed form to each one of these metrics to evaluate the extraction of the narrative elements. In strict form, all the tokens of the narrative element should be labeled by the automatic annotator. For instance, if the human annotator labels a participant "arma de fogo"(fire gun), then a true positive only occurs if the automatic annotator labels "arma de fogo"(fire gun) as well. In the relaxed form, if there is an overlap between the human annotation and the automatic annotation, then we compute it as a true positive. For instance, if the human annotator labels a participant "arma de fogo"(fire gun), then a true positive occurs if the automatic annotator labels "arma"(gun). Although UzZaman et al. employed the relaxed version only for time expressions, we consider the relaxed version of precision, recall, and f1 a pragmatic way to evaluate the results of an automatic labeling framework for spans of text. The reason for this is that the partial match will be highlighted to the human annotator, who can locate more quickly the narrative element associated with the highlighted excerpt. Also, the meaning usually can be understood when there are partial overlaps between the span of entities, and to understand the meaning of narrative is the ultimate goal of the narrative extraction task.

Discussion of results. The results of our experiments are described in Table 2. It is possible to observe that time presents the highest performance among all entities. If we consider the experiments with text2story as being similar to zero-shot annotators, then they yield competitive results in this context. Our baseline results are even competitive with the GPT-3 model for the time extraction.

Concerning participant entities, the performance of SRL is poor across all datasets. Nonetheless, we can notice that the precision of the participants, in all the datasets, is more expressive than the recall. Hence, the SRL is correct when labeling participants, but fails to identify most of them. This happens because of how the output of SRL is treated inside the pipeline. SRL can return frames associated with a verb. The returned frames can span across several tokens, therefore a heuristic is required to decide which frame should be identified as a participant and which is not. If the lexical head of the frame is undefined, the heuristic discards the frame. This elimination likely affects participant recall. When considering the participant entity, the Spacy framework performance was superior to the SRL results, but it is still not competitive with the GPT-3 model.

The results of events in the zero-shot scenario for Lusa News showed relaxed f1 scores above 0.5. Considering that most event triggers are verbs, this is an expected result. However, in the ACE dataset, the numbers are low. One possible reason is that the length of the texts of this dataset is longer, which can harm the performance. Nonetheless, the results of SRL are still low, which can be an indication that the definition of event employed in the labeling of the dataset is not standardized as the Lusa News, which employs an ISO standard in its definition of event. Likely, achieving higher performance in event detection within the ACE dataset requires a fine-tuned model.

5. Conclusion

In this paper, we introduced the text2story Python package, which simplifies narrative extraction from text. It streamlines the use of off-the-shelf libraries and offers an extensible framework for annotation, reading, and visualization. Notably, the visualization module includes the Discourse Representation Structure (DRS), enhancing entity relation inferences.

We also present the results of our package. The first set includes visualizations that facilitate manual and automatic annotation inspection. The second set provides quantitative results from experiments in two datasets: an English dataset for event detection, which contributes to text2story component benchmarking, and a Portuguese dataset. These datasets serve as suitable baselines and support narrative extraction research benchmarks.

We would like to acknowledge some limitations of our package. Firstly, it utilizes transformers, such as BERT, in some of its components; however, it does not currently incorporate the latest Language Models (LLMs). Addressing this limitation is part of our future work. Secondly, our visualization feature is currently available only in a file format. Offering a more interactive visualization option would enhance the annotation inspection process. Third, the definition of the narrative components, although employs a worldly standard, can not encompass several other possible definitions of such elements. Lastly, it is important to note that prompting is a subjective technique, and the results can vary significantly based on the input provided (Liu et al., 2023). Improvements in the results generated by GPT-3 are possible, especially given ongoing discussions about prompt standardization. Nonetheless, our experiment's prompt guidelines are open for examination and customization.

6. Acknowledgments

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020. DOI 10.54499/LA/P/0063/2020 | https://doi.org/10.54499/LA/P/0063/2020 The author also would like to acknowledge the project StorySense, with reference 2022.09312.PTDC (DOI 10.54499/2022.09312.PTDC).

7. Bibliographical References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Thomas Bögel, Michael Gertz, Evelyn Gius, Janina Jacke, Jan Christoph Meister, Marco Petris, Jannik Strötgen, Ryan Cordell, and Anne Baillot. 2015. Collaborative text annotation meets machine learning: heurecléa, a digital heuristic of narrative. *DHCommons journal*, 1.

Johan Bos, Valerio Basile, Kilian Evang, Noortje J Venhuizen, and Johannes Bjerva. 2017. The groningen meaning bank. In *Handbook of linguistic annotation*, pages 463–496. Springer.

Ricardo Campos, Alípio Jorge, Adam Jatowt, Sumit Bhatia, and Marina Litvak. 2023. The 6th international workshop on narrative extraction from texts: Text2story 2023. In *Advances in Information Retrieval*, pages 377–383, Cham. Springer Nature Switzerland.

Ricardo Campos, Alípio Mário Jorge, Adam Jatowt, Sumit Bhatia, and Mark A. Finlayson, editors. 2021. *Proceedings of Text2Story - Fourth Workshop on Narrative Extraction From Texts held in conjunction with the 43rd European Conference on Information Retrieval (ECIR 2021), Lucca, Italy, April 1, 2021 (online event due to Covid-19 outbreak)*, volume 2860 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *LREC*, pages 4545–4552.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Mahmoud El-Haj, Paul Rayson, and Nadhem Zmandar. 2022. Proceedings of the 4th financial narrative processing workshop@ lrec2022. In *Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Bart Geurts, David I. Beaver, and Emar Maier. 2020. Discourse Representation Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, spring 2020 edition. Metaphysics Research Lab, Stanford University.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. 2020. spacy: Industrial-strength natural language processing in python.

Jan Horstmann. 2020. Undogmatic literary annotation with catma. *Annotations in Scholarly Editions and Research*, pages 157–176.

ISO-24617-9. 2019. Language resource management- Semantic annotation framework (SemAF) - - Part 9: Reference annotation framework (RAF). Standard, International Organization for Standardization, Geneva, CH.

Zhuoran Jin, Yubo Chen, Dianbo Sui, Chenhao Wang, Zhipeng Xue, and Jun Zhao. 2021. CogIE: An information extraction toolkit for bridging texts and CogNet. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 92–98, Online. Association for Computational Linguistics.

Prateek Jindal and Dan Roth. 2013. Extraction of events and temporal expressions from clinical narratives. *Journal of Biomedical Informatics*, 46:S13 – S19. 2012 i2b2 NLP Challenge on Temporal Relations in Clinical Data.

Hans Kamp and Uwe Reyle. 1993. *Introduction to Model Theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 42. Springer Netherlands.

Paul R Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*, pages 1989–1993.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Ines Montani and Matthew Honnibal. 2018. Prodigy: A new annotation tool for radically efficient machine teaching. *Artificial Intelligence to appear*.

Sofia Oliveira, Daniel Loureiro, and Alípio Jorge. 2021. Improving portuguese semantic role labeling with transformers and transfer learning. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–9.

Details omitted for double-blind reviewing. 2021. tool.

Details omitted for double-blind reviewing. 2023. Text2story lusa annotated.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Maria da Purificação Silvano, Evelin Amorim, António Leal, Inês Cantante, Maria de Fátima Henriques da Silva, Alípio Jorge, Ricardo Campos, and Sérgio Sobral Nunes. 2023. Annotation and visualisation of reporting events in textual narratives. In *Proceedings of Text2Story 2023: Sixth Workshop on Narrative Extraction From Texts*.

Purificação Silvano, António Leal, Fátima Silva, Inês Cantante, Fátima Oliveira, and Alípio Mário Jorge. 2021. Developing a multilayer semantic annotation scheme based on iso standards for the visualization of a newswire corpus. In *Proceedings of the 17th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 1–13.

H. Sousa, R. Campos, and A. Jorge. 2023a. Tei2go: A multilingual approach for fast temporal expression identification. In *32nd ACM International Conference on Information and Knowledge Management*, Birmingham, United Kingdom.

Hugo Sousa, Nuno Guimarães, Alípio Jorge, and Ricardo Campos. 2023b. Gpt struct me: Probing gpt models on narrative entity extraction. In *The 22nd IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology*, WI = Artificial Intelligence in the Connected World.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Michael Stewart, Wei Liu, and Rachel Cardell-Oliver. 2019. Redcoat: A collaborative annotation tool for hierarchical entity typing. In *Proceedings of the 2019 Conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP): system demonstrations*, pages 193–198.

Jannik Strötgen and Michael Gertz. 2010. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 321–324.

Michael Toolan. 2012. *Narrative: A critical linguistic introduction*. Routledge.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9.

Chao Wang, Rongxin Wu, Haohao Song, Jiwu Shu, and Guoqing Li. 2022. smartpip: A smart approach to resolving python dependency conflict issues. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–12.

Ying Wang, Ming Wen, Yepang Liu, Yibo Wang, Zhenming Li, Chao Wang, Hai Yu, Shing-Chi Cheung, Chang Xu, and Zhiliang Zhu. 2020. Watchman: Monitoring dependency conflicts for python library ecosystem. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 125–135.

Haoyang Wen, Ying Lin, Tuan Lai, Xiaoman Pan, Sha Li, Xudong Lin, Ben Zhou, Manling Li, Haoyu Wang, Hongming Zhang, Xiaodong Yu, Alexander Dong, Zhenhailong Wang, Yi Fung, Piyush Mishra, Qing Lyu, Dídac Surís, Brian Chen, Susan Windisch Brown, Martha Palmer, Chris Callison-Burch, Carl Vondrick, Jiawei Han, Dan Roth, Shih-Fu Chang, and Heng Ji. 2021. RESIN: A dockerized schema-guided cross-document cross-lingual cross-media information extraction and event tracking system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 133–143, Online. Association for Computational Linguistics.

Ningyu Zhang, Xin Xu, Liankuan Tao, Haiyang Yu, Hongbin Ye, Shuofei Qiao, Xin Xie, Xiang Chen, Zhoubo Li, and Lei Li. 2022. DeepKE: A deep learning based knowledge extraction toolkit for knowledge base population. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 98–108, Abu Dhabi, UAE. Association for Computational Linguistics.

A. Full Manually Annotated Example

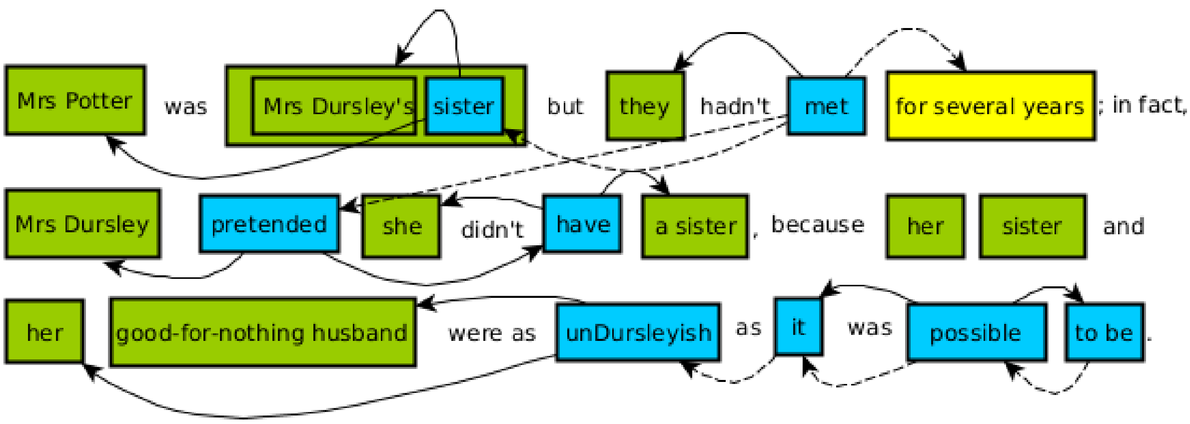


Figure 7: The Example 1 annotated by a human. The dashed lines are temporal links and the solid lines are semantic links.