University of Amsterdam
Informatics Institute

Timeline Extraction using ChatGPT

from Decision Letters

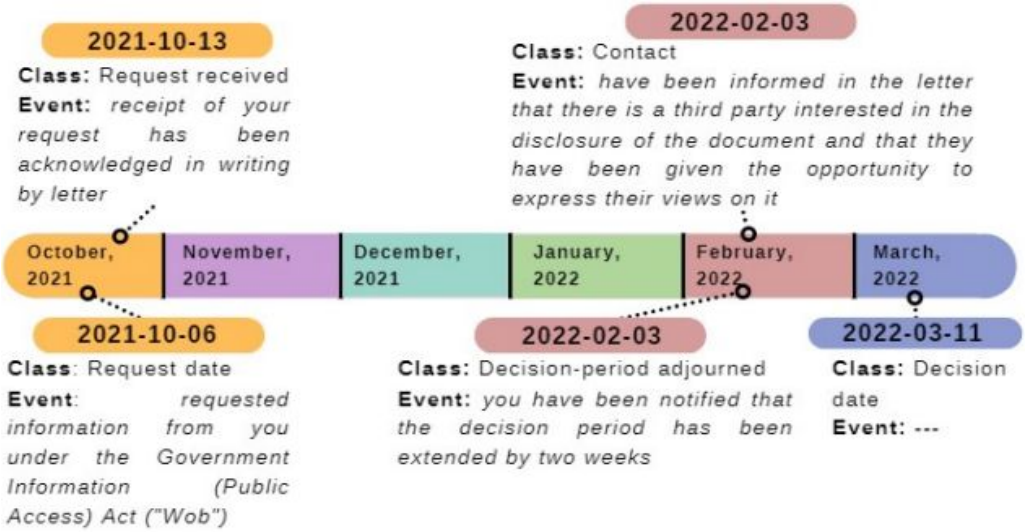
CASE @ EACL 2024

Femke Bakker, Ruben Van Heusden & Maarten Marx

Research focus - Timeline Extraction

Timeline extraction:

- Input: document
- All dates with events
- Ordered on the timeline



Research focus - Event Extraction

Event extraction:

- Input: sentence
- Output: triple(s):
  - Date
  - Event phrase
  - Event class

Research focus - Event Extraction

Event extraction:

- Input: sentence
- Output: triple(s):
  - Date
  - Event phrase
  - Event class

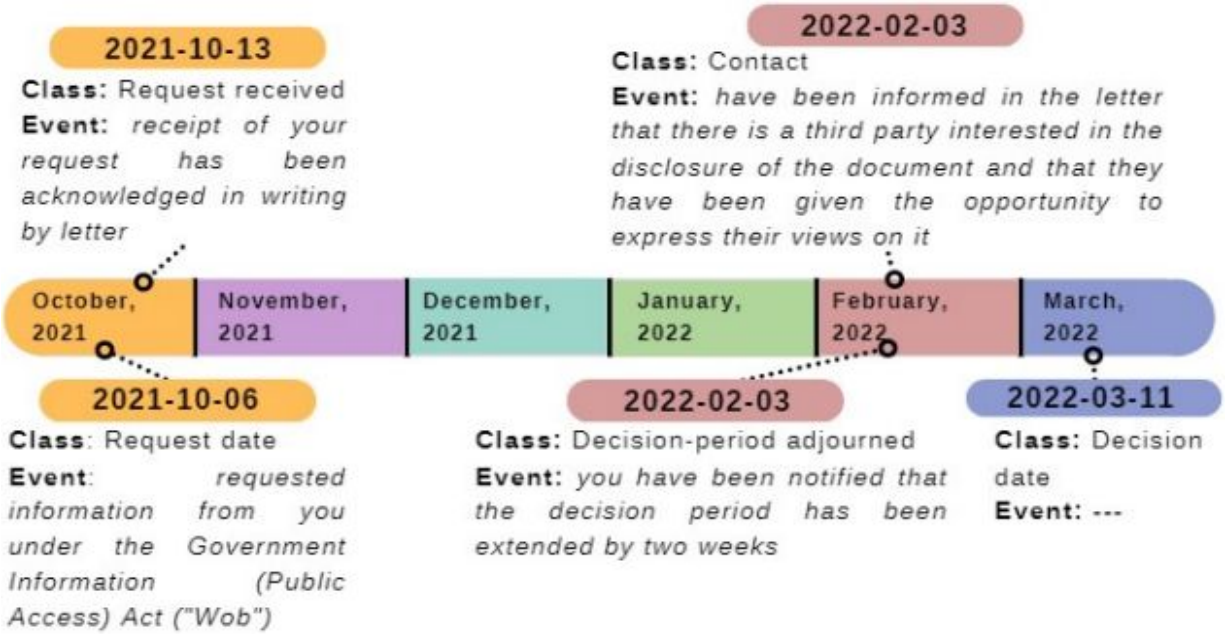
Event phrase

"In your email of 3 August 2022, you requested my ministry to provide additional information"

Date

Event class: request date

Research focus - Timeline Extraction



Relevance

Applications of timelines:

- Graphical summaries
- Process mining
- Process monitoring

Related work

Traditional Methods:

- Pipeline approaches:
  - Part-of-Speech tagging
  - Named Entity Recognition (NER)
  - Coreference resolution (relation entities and events)
- Problem: propagation of errors

Du and Cardie (2020):

- BERT -> Question Answering task to extract event phrases.
- Similar to our approach

Open government act (Woo)

- Open government act -> the right to information about the government.

(Open state foundation, 2023)

Dataset

- 100 Dutch (Woo) decision letters
  - originating from Dutch ministries.
- SpaCy extracts sentences with dates
- 812 annotated sentences containing dates
- 4 subtasks during annotation:
  - Correct dates
  - Label event phrase
  - Classify event (total of 8 classes)
  - Label relation between date and event phrase
- 50/50 development/test set

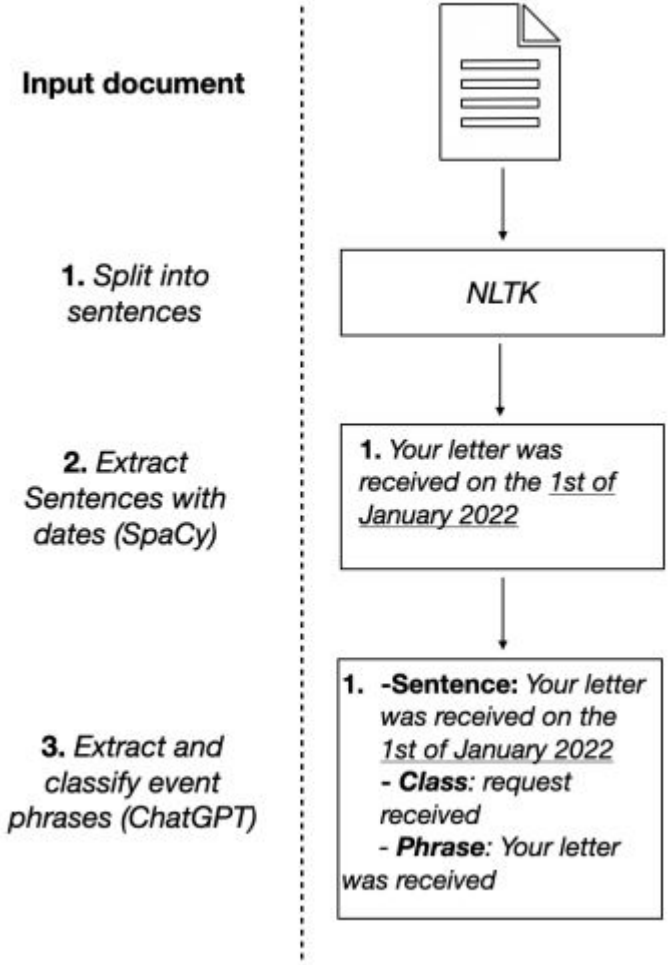
Timeline Extraction Pipeline

Step 1: Sentence splitting

Step 2: Extract sentences containing dates

- using SpaCy
- many false positives -> filter
- remove sentences without dates

Step 3: Extract event phrases and classes using ChatGPT



Step 3: Extract event phrases and classes using ChatGPT

Event phrase extraction

Prompt 1

Input:
sentence with list of dates in sentence.

Instruction:
return event phrase of the dates
OR return 'no event' if date has no event

Examples:
5 most similar sentences + ground truth event phrase (based on BM25)

Event classification

Prompt 2

Input:
list of event phrases -> output from prompt 1

Instruction:
classify event phrases
+ description of each class

Examples:
2 examples of similar event phrases + ground truth class (based on BM25)

Result: triples - {date, event phrase, class)

SpaCy Prompt 1 Prompt 2

Results - Date extraction

Using SpaCy

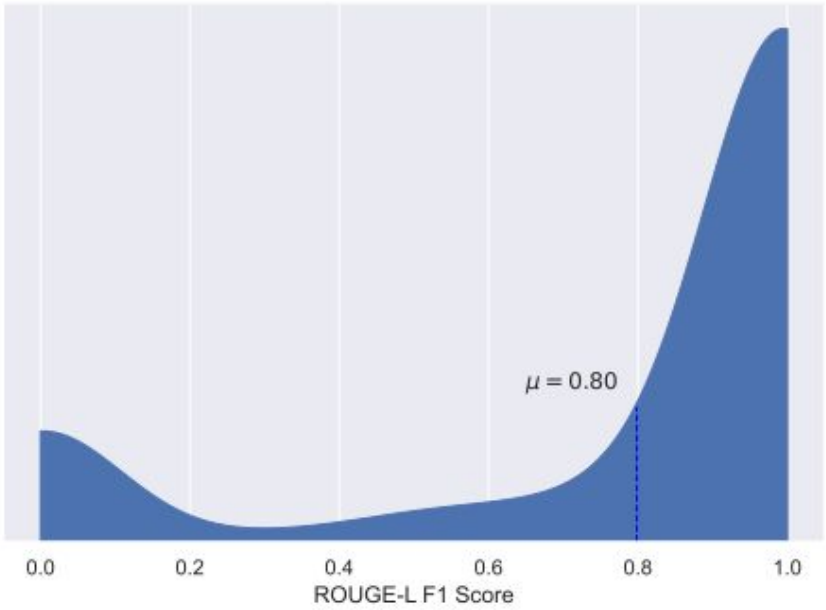
- Accuracy = 94%
- Mistakes: ambiguity in the date
- Example: “In the month of June...”

Evaluation - Event Phrase Extraction

ROUGE-L metric

- Match longest subsequence of words

Results - Event extraction



- ROUGE-L threshold at 0.5
- Accuracy = 82%
- Precision = 83%
- Recall = 78%

Mistakes:

- Dates without event phrase

Results - Event classification

	Precision	Recall	F1-score	Support
Decision period adjourned	0.93	0.93	0.93	29
Contact	0.89	0.79	0.84	42
WOO legislation in effect	1.00	1.00	1.00	16
Confirmation request received	1.00	0.98	0.99	44
Other	0.73	0.67	0.70	24
Request date	0.98	0.98	0.98	48
Request received	0.75	1.00	0.86	15

ChatGPT classification:

- Overall Macro F1-score = 79%
- Worst performing class: Other
- Performance best -> event phrases similar patterns

Results - Timeline Construction

76%

completely correct triples
(out of 524)

20%

completely correct timeline
(out of 50)

- Average timeline contains 8 dates
- Mistakes per timeline: mode=1 & average=3.2
- Filter obvious mistakes

Conclusion

- Little training data needed for quite accurate timeline extraction using ChatGPT

- Future work: open-source -> Llama-2

Timeline Extraction from Decision Letters Using ChatGPT

Bakker, F., Van Heusden, R., & Marx, M. (2024, March). Timeline extraction from decision letters using chatgpt. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)* (pp. 24-31).

University of Amsterdam
Informatics Institute

GitHub