The Knowledge Engineering Review, Vol. 00:0, 1–24. © 2004, Cambridge University Press
DOI: 10.1017/S000000000000000 Printed in the United Kingdom

TempCourt: Evaluation of Temporal Taggers on a new Corpus of Court Decisions

María Navas-Loro, Erwin Filtz, Víctor Rodríguez-Doncel, Axel Polleres and Sabrina Kirrane[1]

María Navas-Loro and Víctor Rodríguez-Doncel
Universidad Politécnica de Madrid, Montegancedo Campus
D3204 – Ontology Engineering Group
Madrid, Spain
E-mail: {mnavas, vrodriguez}@fi.upm.es
ORCiD: 0000-0003-1011-5023, 0000-0003-1076-2511

Erwin Filtz, Axel Polleres and Sabrina Kirrane
Vienna University of Economics and Business
Institute for Information Business
Vienna, Austria
E-mail: {firstname.lastname}@wu.ac.at
ORCiD: 0000-0003-3445-0504, 0000-0001-5670-1146, 0000-0002-6955-7718

Abstract

The extraction and processing of temporal expressions in textual documents has been extensively studied in several domains, however for the legal domain it remains an open challenge. This is possibly due to the scarcity of corpora in the domain and the particularities found in legal documents that are highlighted in this paper. Considering the pivotal role played by temporal information when it comes to analyzing legal cases, this paper presents TempCourt, a corpus of 30 legal documents from the European Court of Human Rights, the European Court of Justice and the United States Supreme Court with manually annotated temporal expressions. The corpus contains two different temporal annotation sets that adhere to the TimeML standard, the first one capturing all temporal expressions and the second dedicated to temporal expressions that are relevant for the case under judgment (thus excluding dates of previous court decisions). The proposed gold standards are subsequently used to compare ten state-of-the-art cross-domain temporal taggers, and to identify not only the limitations of cross-domain temporal taggers but also limitations of the TimeML standard when applied to legal documents. Finally, the paper identifies the need for dedicated resources and the adaptation of existing tools, and specific annotation guidelines that can be adapted to different types of legal documents.

Keywords: legal corpus, temporal annotation, case law, legal NLP, evaluation

1 Introduction

Legal information systems are indispensable tools for many legal practitioners. An emerging area of research is the use of text analytics to derive structured data from legal text (e.g. norms, opinions, recommendations or court decisions). In this context, one of the most relevant activities is the automatic extraction and processing of events and temporal expressions with a view to creating timelines.

In this context, a *temporal expression* (TE) is a word or sequence of words making reference to a time instant (e.g. 'seven o'clock') or a time interval (e.g. 'from seven to ten'). Temporal expressions frame events or happenings implicitly or explicitly mentioned in the document. *Temporal relations* bind TEs to events and determine the relative position of some events with respect to other events (through relations such as 'after' or 'before').

The example below is a text excerpt from a court decision of the European Court of Human Rights describing the facts of the *Aras v. Turkey* case (no. 21824/07, 20 July 2017). The text contains three TEs

[1] The two first authors equally contributed to this work.

(in bold below), two of them being in an absolute form (e.g. *11 December 2002*) and one in a relative form (*same day*).

> ”On **11 December 2002** the applicant’s statement was taken by the public prosecutor and, on the **same day**, the judge at *Istanbul State Security Court* ordered her detention on remand. On **7 December 2002** the applicant was arrested on suspicion of membership of a terrorist organisation.”

This temporal information is related to three events, namely, the public prosecutor taking the statement, the judge ordering a detention, and the applicant being arrested. Each of the events is related to the other entities, either named (*Istanbul State Security Court*) or not (the applicant). Although the two absolute dates in the text above appear in the same format, this is not always the case and very often different formats are used even within the same document. Although our exemplary legal case can be used to motivate an investigation into both temporal and event extraction (e.g. (39, 46)), in this paper we focus specifically on temporal expressions.

Temporal taggers operate on texts like the one above, performing different tasks, namely TE *identification*, *normalization*, and *classification*. *Identification* (also called *detection* or *extraction*) is a task which involves finding TEs and their start and end position in the text. *Normalization* (or *anchoring*) is a task that interprets TEs to obtain specific instants and intervals represented in a standard format. This task resolves relative TEs (as ‘the same day’) from context information, localizes time formats (i.e. mm/dd/yy vs dd/mm/yy), considers timezones and enables the reformatting of the TEs into a standard format (e.g. ISO 8601[2]). In contrast, a *classification* task is used to determine which kind of TEs have been found. For instance, **on 7 December 2002** is most likely a time point, while **from 7 December 2002 to 12 December 2002** is a time interval. The temporal expressions found by the temporal taggers are usually represented in domain-agnostic formats, such as TimeML[3]. TimeML is the most widely accepted mark-up language for temporal expressions, and its use is justified over domain-specific formats (e.g. Akoma Ntoso[4] in the legal domain) for it permits representing more details and nuances specific to the temporal terms.

Although several temporal taggers have been proposed and investigated in different domains, the suitability of existing methods to extract temporal information from legal texts has been relatively unexplored to date as being only a side effect for other tasks, for instance document classification or reasoning over documents. Additionally, the lack of temporal resources in the domain is a major drawback when it comes to research in this direction. To the best of the authors’ knowledge, there is no preexisting temporal annotation gold standard based on legal text corpora. Consequently, there is no previous evaluation of how well standard temporal tagging tools perform in this domain. To this end, this paper makes the following contributions:

- an analysis of the particularities of temporal annotation in the legal domain;
- the provision of a temporally tagged corpus (named TempCourt, freely available online[5]), composed of legal documents from three sources, namely the European Court of Human Rights, the European Court of Justice and the United States Supreme Court; and
- a broad comparison of state-of-the-art cross-domain temporal taggers using the proposed gold standard.

The remainder of this paper is structured as follows: Section 2 describes existing work on temporal information extraction. Section 3 examines the particularities of dealing with temporal expressions in the legal domain. Section 4 presents the methodology used for the construction of the TempCourt corpus. Section 5 introduces several existing temporal taggers. Section 6 evaluates ten state-of-the-art temporal taggers over documents from three different legal sources, namely the European Court of Human Rights,

[2] `https://www.iso.org/iso-8601-date-and-time-format.html`
[3] `http://www.timeml.org`
[4] `http://www.akomantoso.org/`
[5] `https://tempcourt.github.io/TempCourt/`

the European Court of Justice and the United States Supreme Court. Finally, Section 7 presents our conclusions and discusses future work.

2 Related Work

Temporal tagging is a mature area of research that has been applied in different contexts, but scarcely in the legal domain. This section reviews several corpora with temporal annotations, along with the work done previously in temporal annotation of legal texts and in other domains.

The temporal information of a text document can be represented in structured, ad-hoc formats such as TIDES TIMEX2 (10) or TimeML (35). TimeML is the ISO standard[6] for time and event markup and annotation. Other general-purpose annotation standards can also be used to represent TEs, such as the W3C Web Annotations[7] or the NLP Interchange Format[8] (NIF) (15). TimeML uses TIMEX3 tags (modelled on previously mentioned TIMEX2) for marking TEs, and distinguishes between different `types` (namely, `DATE, DURATION, TIME` and `SET`, the latter being the type associated with sets of recurrent times). Other attributes in TIMEX3 tags allows for the expression of temporal information as a normalized value, for instance the actual date instead of relative expressions such as `yesterday`, following the ISO 8601 standard (`value`). TIMEX3 can also mark the presence of modifiers (`mod`) such as `END` or `LESS_THAN`, or specific information for each type, such as the frequency (`freq`) for `SET`.

Thus, for the analysis of temporal expressions, the following three domains received the most attention: medical texts (e.g. the THYME corpus (44)), news (e.g. the Timebank corpus (34) and the MEANTIME corpus (29)) and historical documents (e.g. the Wikiwars corpus (28)). Corpora have also included texts in different language registers, such as tweets (45), colloquial texts (43) or scientific abstracts (43). However, to the best of the authors' knowledge, there are no temporally annotated legal corpora publicly available that relate to English language court decisions. Although annotation challenges (both in general and also in different specific domains) have been identified in literature (17, 43, 44), very little work has been conducted in connection with the legal domain. A description of the different approaches adopted by existing temporal taggers, including the identification of several state-of-the-art temporal taggers, can be found in Section 5.

In the legal domain, previous research work by Schilder (39) already pointed out the relevance of the temporal dimension of information in legal documents. In this work, an analysis of the different types of legal documents and the temporal information that can be found in them was outlined. Schilder distinguished between dates in transactional documents (namely, documents written by lawyers for specific transactions, such as contracts or agreements), constraints in statutes or regulations, and legal narratives in case law. While the first two types of documents received dedicated attention, narratives in case law were assimilated to narratives present in news. An alternative approach proposed by Isemann et al. (16), used both Named Entity Recognition (NER) and temporal processing to extract temporal dependencies from regulations with no narrative-structure. The authors also described some of the recurrent pitfalls temporal taggers have to deal with, such as the confusion between legal references (e.g. 'Directive *2009*/28/EC') and actual dates, as shown in Table 2, or the distinction between *episodic* and *generic* events —the former referring to a specific moment (e.g. 'the rescission of the contract was done on 7 December 2017') and the latter referring to an event in general truths, laws, rules or expectations (e.g. 'Every rescission implies the following actions'). Other approaches in the legal domain include works on transactional documents by Naik et al. (30), where a first framework for dealing with temporal information in that kind of texts is proposed. Also additional efforts focused on reasoning with legal evidence (burden of proof) and coherence of narratives (e.g. plausibility and completeness) were made (49), using temporal information but without extracting it from scratch.

[6] ISO 24617-1 Language Resource Management - Semantic Annotation Framework (SemAF) - Time and Events (SemAF Time and ISO-TimeML)

[7] `https://www.w3.org/TR/annotation-model/`

[8] `http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#`

Works in other fields, such as the medical domain, are also of interest since they share common requirements, such as the need of domain knowledge for identifying specific events[9] and for dealing with the existence of several timelines in the same text, among others. The analysis by Styler et al. (44) in the clinical domain identifies the need of specific guidelines for temporal annotation, which require domain-specific temporal knowledge and the definition of general phases in clinical processes (some kind of commonsense domain knowledge). Furthermore, new tags not included in the temporal annotation standard TimeML, commonsense information and events are defined in the same work, along with annotation needs and different timelines (such as discussions with other colleagues and notes about risks in treatment) were redesigned for fitting the medical particularities. We work under the assumption that most of these considerations and challenges can also be present in a similar form in legal documents, requiring therefore a dedicated approach. We conclude that one of the primary limitations of existing work is the fact that no special consideration has to date been given to both the narrative structure and the particularities of the legal domain (see Section 3 for additional details).

3 Particularities of Legal English

Temporal information is a very important aspect of legal cases. It has an effect on the version of the applicable law and it creates a chronological order of events in a legal case. Sometimes it is important to know whether event *A* or event *B* happened first. In addition, temporal information is also used to assess whether past events may be time-barred.

When it comes to the automatic extraction of temporal information from legal documents, it is important to highlight that legal documents, and in particular court decisions, slightly differ in structure and writing style from documents from other domains. These differences include deeper parse trees, differences in part-of-speech distributions and more words per sentence (8).

Judgments are usually framed in legal processes following specific procedures, events and timings, whose mention in the judgment constitutes context information that should not be lost in the annotation process. An example of this is the concept of *preliminary ruling*, a legal term referring to a phase previous to the decision when the European Court of Justice is asked how law should be interpreted, being therefore a reference to this period and a hint for temporal localization of other events. Also specific events happening in legal frameworks must be considered when processing legal texts, as done in other domains such as the medical domain (44).

3.1 Structure of Judgments

Table 1 illustrates the differences in document structure for judgments made by the European Court of Justice (ECJ) and the United States Supreme Court (USC), and preliminary assessments of applications submitted to the European Court for Human Rights (ECHR). The court decisions from the European courts follow a similar structure that already hints which categories of TE could be expected in different parts of the texts. In particular, both ECJ and ECHR start with a description of the involved parties (section A) and are then followed by a case summary (B), stating concisely why this case has been brought to the respective court and what happened so far in terms of the legal proceedings. In ECJ decisions, the legal proceedings are followed by the applicable legal framework and then by the case description, whereas the ECHR structure is the other way round. The decisions of the ECJ and ECHR courts conclude with the matching of the law with the facts of the case under the legal basis (E) and the resulting judgment (F). In contrast to ECHR documents, the ‘Legal framework’ section (D) in ECJ documents cites European and local legislation, without any direct references to the case itself, and as such this information was excluded from the final documents in the corpus presented in this paper. Although TEs corresponding to other related events such as prior decisions could be extracted from these sections, we focus on case-related temporal information and leave the extraction of events for future work. Apart from beginning with the involved

[9] For instance, *diagnosis* such as *tumors* or medical tests are relevant events that should appear in a timeline of a medical doctor, as stated by Styler et al. (44), but not in other types of texts. Similarly, specific legal events such as *preliminary rulings* (explained in Section 3) in European judgments are always relevant to lawyers, although they never appear in other kinds of texts.

Table 1 Structure of ECJ, ECHR and USC decisions.

Section	ECJ	ECHR	USC
A	Involved parties	Involved parties	Involved parties
B	Case summary	Procedure	Syllabus
C	Legal framework	Circumstances of the case	Main Opinion
D	Circumstances of the case	Legal framework	Concurring and dissenting opinions
E	Court assessment	Court assessment	
F	Judgment	Judgment	

parties (A) in a particular case, the structure of USC decisions is quite different. The second section (B) is called ‘syllabus’ and contains a short summary of the case. It is followed by the main opinion (C), that includes the final decision of the court and explains how the court came to this decision, by referring to the legal foundations. The last part of a decision states, where applicable, the concurring and dissenting opinions of the involved justices (D). An opinion is called ‘concurring’ if a justice follows the main opinion but grounds the decision on a different legal rationale. A dissenting opinion is issued in cases where a justice disagrees with the main opinion and the underlying legal rationale. Following a consistent structure makes legal documents comparable, and fulfills the expectations of readers who are used to find a specific kind of information always at the same place in the same kind of legal document. Furthermore, the consistent structure of legal documents (from the same authority or within a jurisdiction) leads to expectations with respect to the type of temporal information which could be expected in each section of the document. We expect temporal references describing the facts of the case (*what happened when?*), which could be used for generating timelines for document summarization, to be present in the *case summary* (ECHR), *Circumstances of the case* (ECJ) or *syllabus* (USC) sections in the judgment of the respective deciding court, but mentions to general temporal events to appear throughout the entire document. The structural properties of legal documents could also be exploited for the automatic creation of timelines as legal documents can be very long. For the analysis of a judgment it is necessary to understand the order of the events as this can affect the legal proceedings. The easier understanding could be supported with a visual representation of the order of events, hence a timeline that shows the important events and provides a visual summary of the case.

Dates are used in virtually every domain. In contrast to posts published in social media, e.g. Facebook or Twitter, where every user might write dates in different formats, documents from official authorities, such as courts, usually use the same format to represent dates in all documents. Differences in date representation that can be noticed are for instance the order of day and month or the used separators. Therefore, the differences in date representation are seldom found within a document, but may vary from court to court.

3.2 Mistaken or Misleading Temporal Expressions in Legal Documents

References to legal documents often include some sort of temporal information, usually forming a text pattern prone to be confused with a true temporal expression (see examples in Table 2). Typical references containing temporal information are references to previous court cases, laws or legal literature, where the temporal information indicates a point in time when the respective reference has been decided or published. However, temporal information contained in references is not considered relevant for a specific case in terms of describing *what happened when?*. For example, the expressions in Table 2 convey some temporal information, e.g., four-digit sequences that could be recognized as years, but which only in some cases indeed refer to actual years. Tagging these kind of expressions as TEs may become a major problem and lead to additional errors —for instance nearby dates in the text can be normalised from these wrong references leading to further errors in the entire text. Additionally, references to other legal documents often present their creation date, that must be differentiated from dates in the document timeline of referred case events. An example of this, where the given date refers to the date of a Council Directive of the European Union and thus is irrelevant for the narrative of the text, is the excerpt below:

Table 2 Examples of mistaken and misleading temporal information.

Source	Example	Description
ECHR	no. 7334/13, 127 - 128, ECHR **2016**	Reference to another case
ECHR	Timoshin v. Russia (**dec.**)	Reference to a decision (dec.), often confused with the month of December
ECJ	OJ **2008** L 348 p. 98	Reference to official journal of the EU
ECJ	Directive **2008**/115/EC	Reference to a directive published in 2008
USC	See Va. Code Ann. §53.1-165.1 (**2013**)	Law reference
USC	[...] 772 F. 3d **1328**, **1333** (CA10 **2014**)	Precedent case reference

> ”Council Directive 93/13/EEC of **5 April 1993** on unfair terms in consumer contracts must be interpreted as not precluding (...)”

For processing these kinds of expressions, we could first detect and hide them from the temporal tagger (e.g., replacing them for an innocuous expression before the processing and restoring them afterwards) or alternatively we could filter them in a post-processing step.

3.3 Incompleteness of the TimeML Standard for the Annotation of Legal Documents

During the annotation of the corpus presented in this paper, we also detected relevant information that the TimeML standard is not able to represent. The main drawbacks of the TimeML standard applied to legal documents are summarized in the following subsections.

3.3.1 Specific Legal Terminology as Modifiers

Documents in the legal domain are rich in non-colloquial noun phrases representing temporal information. For example, the sentence “the *expiry* of the three-day period” is badly understood by parsers in comparison with “the *end* of the three-day period”.

Similarly, when the extension of a duration is uncertain (e.g, range between two points, such as in “*period of between seven and thirty days*”), there is no way to properly represent the uncertainty. Likewise, when referring to different possibilities frequently found in the legal language such as “*was a year or two more of prison time*”, this information cannot be properly annotated —even if some taggers such as SUTime (4) provide alternative values for similar expressions, i.e. “*from one to two years*”, the standard specification does not allow them.

The standard should be able to represent all these particularities of the legal domain. Similarly, a temporal tagger for the legal domain should be able to reason with this level of granularity.

3.3.2 Missing Levels of Granularity

Temporal expressions are important in the legal domain. Not only points in time which are used to determine the applicability of a particular law, but also durations are of high importance especially in formal laws determining the limitations of time (e.g. to plead the statute of limitations) for actions that must be taken before they preclude. For instance, in the legal domain a different way to count days is often applied. While `DURATION` is sufficient to indicate the absolute lapse of time, TimeML is not sufficient to indicate non-absolute durations such as “*10 working days*”.

Temporal taggers could be enhanced with external knowledge to recognize special constraints being applied to durations, for instance, work calendars where working days are identified. Eventually, also the capability to reason at this level of granularity would be desirable.

3.3.3 Exhaustive List of Attributes

The TimeML attribute `functionInDocument` allows for the marking of some temporal expressions as special reference ones, but just as one among: *‘creation_time’*, *‘expiration_time’*, *‘modification_time’*, *'publication_time'*, *'release_time'*, *'reception_time'* or *'none'*. This is not enough for legal documents, where domain expressions such as *'lodgement_time'*, *'argued_time'* or *'decision_time'* would be more useful. Domain-specific extensions to the TimeML standard could be used to solve this particular problem.

3.3.4 Limited Expressivity of the Existing Format

There are temporal expressions whose anchor time is not the DCT (Document Creation Time) nor are they related to any temporal expressions in the text, but in other legal documents cited in the text, such as in "*The dissent also relies heavily on Missouri v. Frye, 566 U. S. 134 (2012), and Lafler v. Cooper, 566 U. S. 156 (2012). (...) Lafler, decided* ***the same day*** *as Frye (...)*".

To cover this issue a temporal tagger needs to be combined with a co-reference system in order to find the matching events to which a certain temporal expression relates. This could be addressed by making use of the clear structure of legal documents which usually use the same citation style in all documents such that temporal expressions appearing next to case references can be annotated as belonging to them.

The official TIMEX3 tags cannot properly represent precise intervals on their own. A time interval such as "*between 12.45 and 18.45*" can only be represented as a DURATION (of 4 hours) or as two unrelated datetime points. This is a problem in cases where exact intervals are needed to solve legal problems such as confirming an alibi or evaluating exact timespans.

While some of these limitations could also be found anecdotally in other kinds of texts, they are common in legal documents, and relevant to their temporal dimension. Other non legal issues raised when using the TimeML standard are the correct extent of the tags or how to deal with multiple normalization options (for instance, "*one decade*" can be "`P1DE`" or "`P10Y`", and "*a few weeks later*" can be a duration with a known *beginPoint* or a `FUTURE_REF`).

3.4 Temporal Dimensions

In legal texts temporal expressions can be attributed to different temporal dimensions. We identify three different temporal dimensions and illustrate them based on the example decision *Sophie Mukarubega v Préfet de police and Préfet de la Seine-Saint-Denis* (ECLI:EU:C:2014:2336).

3.4.1 Temporal Dimension of the Legal Process

Each court proceeding is based on some formal rules and new events are added with the gradual advancement of the legal proceeding. This temporal dimension covers events related to the legal process itself, for instance the date a lawsuit has been filed, date of the hearings or the decision date.

> "By a decision of **21 March 2011**, adopted after hearing the person concerned, the Director General of the Office francais de protection des refugies et apatrides (OFPRA) (Office for the protection of refugees and stateless persons) rejected her application for asylum. (...)"

This temporal expression indicates that a certain event has happened, in this case the rejection of asylum.

3.4.2 Temporal Dimension of the Case

This temporal dimension covers factual information about the case which serves as the basis for a judgment.

> "Ms Mukarubega, who was born on 12 March 1986 and is of Rwandan nationality, entered France on **10 September 2009** in possession of a passport bearing a visa. (...)"[10]

The highlighted date refers to a fact of the case, hence the entrance of France.

[10] Please note that the same sentence contains two temporal expressions which are attributed to two different temporal dimensions.

3.4.3 Temporal Dimension of the Legal Context

Temporal information can also affect the legal context and determine the applicable law and the degree of the resulting penalty. This is especially relevant when determining the limitation of liability in time or when checking a legal reference to know the applicable law version. We can illustrate this in the following example of a preliminary ruling request to the European Court of Justice with the dates marked in bold.

> "(...) This request for a preliminary ruling concerns the interpretation of Article 6 of Directive 2008/115/EC of the European Parliament and of the Council of **16 December 2008** (...)"
>
> "Ms Mukarubega, who was born on **12 March 1986** and is of Rwandan nationality, entered France on **10 September 2009** in possession of a passport bearing a visa. (...)"[10]
>
> "By a decision of **21 March 2011**, adopted after hearing the person concerned, the Director General of the Office francais de protection des refugies et apatrides (OFPRA) (Office for the protection of refugees and stateless persons) rejected her application for asylum. (...)"

The first, third and fourth temporal expression refer each to a point in time that is relevant for the legal context. A preliminary ruling for the interpretation of an article requires the article to exist (first date). In the second paragraph, the birth date is general information about the defendant, which does not affect the *temporal dimension of the case* but might influence the *temporal dimension of the legal context*. This is especially important in criminal cases when the birth date in conjunction with the date of the offence constitutes the application of the criminal law relating to juvenile offenders. The third date, on the other hand, refers to a fact of the case, the day of entrance in the host country, being therefore part of the *temporal dimension of the case*. Finally, the fourth date indicates when a decision on the case in the legal process was reached, so this TE corresponds to the *temporal dimension of the legal process*.

3.4.4 Conflict of Temporal Dimensions

One could wonder whether there is the possibility of an overlap of temporal dimensions such that a single event might be part of the *temporal dimension of the legal process* and of the *temporal dimension of the case*. For instance, in cases that go through the entire hierarchy of courts, decisions are reversed by higher courts and referred back to the previous court. In these cases the judgments of the previous courts do have an influence on the following proceedings as courts might be bound to former judgments or receive an order to investigate certain parts of former proceeding in more detail and do more investigation work. However, from our perspective the *temporal dimension of the case* encompass the events inherent to the case, while revisions and case remands do not change anything in the temporal order of events in the original case, instead such information adds context which is relevant for the further proceeding without affecting the *temporal dimension of the case*.

In this section we outlined the particularities of documents in the legal domain which encompass the special structure of judgments, legal terminology, annotation standards such as TimeML and its incompleteness for annotation tasks in the legal domain as well as a classification of temporal dimensions present in judgments.

4 Temporal Annotation

In this section, we aim at evaluating in how far the automatic identification (and normalization) of temporal expressions is feasible using existing taggers, and to test the effectiveness of such tools. In order to enable such an evaluation, we propose two gold standards, one domain focused (LegalTimeML, composed of temporal information important for the facts of the case) and one generic (StandardTimeML, including all temporal information), that can be used to compare the results of temporal taggers and to determine which of them is most suited to be used when working with legal documents. The temporal annotation of all documents used in this work is based on the TimeML annotation language[11]. Figure 1 illustrates the

[11] `https://catalog.ldc.upenn.edu/docs/LDC2006T08/timeml_annguide_1.2.1.pdf`

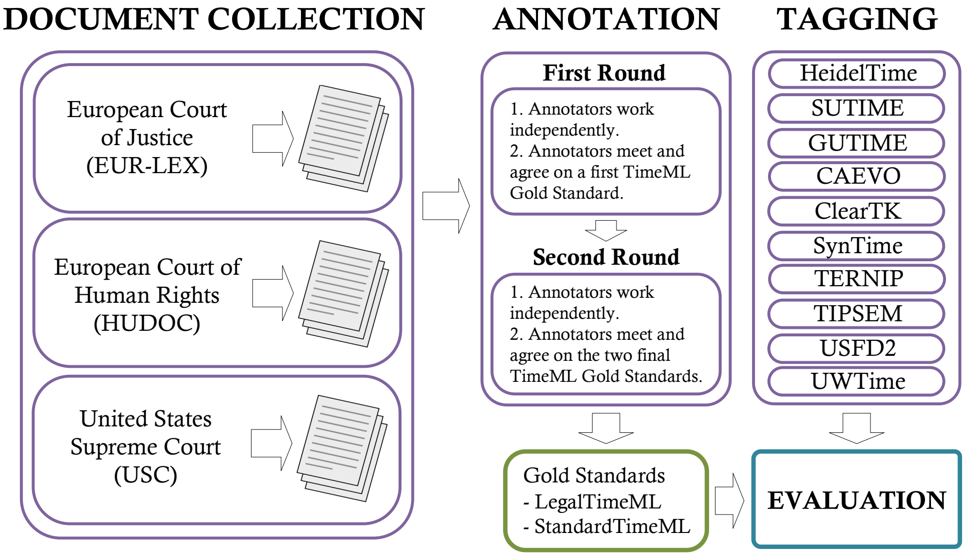
DOCUMENT COLLECTION
ANNOTATION
TAGGING
European Court of Justice (EUR-LEX)
European Court of Human Rights (HUDOC)
United States Supreme Court (USC)
First Round
1. Annotators work independently.
2. Annotators meet and agree on a first TimeML Gold Standard.
Second Round
1. Annotators work independently.
2. Annotators meet and agree on the two final TimeML Gold Standards.
HeidelTime
SUTIME
GUTIME
CAEVO
ClearTK
SynTime
TERNIP
TIPSEM
USFD2
UWTime
Gold Standards
- LegalTimeML
- StandardTimeML
EVALUATION

Figure 1: Outline of our work, including document collection, annotation and evaluation of taggers.

methodology we followed in order to create and evaluate our proposed gold standards. In the document collection phase we retrieve the documents, and in the annotation phase we create in two rounds the gold standards which are then used to compare to the results retrieved from the temporal taggers in the tagging phase.

4.1 Document Collection

Although different types of documents could have been chosen to create a gold standard in the legal domain, our proposed corpus TempCourt is composed of judgments and preliminary assessments of applications as they contain a large number of temporal expressions.

As many of the taggers do not have full support to other languages, we selected court decisions in English to enable a fair comparison of the results of the temporal taggers. Also, in order to increase the variety of ways in which temporal information is represented in different types of courts, we decided to investigate the judgments of courts acting in different jurisdictions and domains. Specifically, we focus on the court decisions of the European Court of Justice (ECJ), which is the highest court of the European Union, and of the United States Supreme Court (USC), and on preliminary assessments of applications submitted to the European Court of Human Rights (ECHR). The documents for the two European courts are available in the respective databases, namely EUR-Lex[12] for the ECJ and HUDOC[13] for the ECHR, while the USC documents were collected from the website of the United States Supreme Court[14]. The corpus created for this work, named TempCourt, consists of thirty court decisions, composed of an even distribution of ten documents per court in each subcorpus. Documents provided by the European Court of Human Rights are allowed to be reproduced for private use or for the purposes of information and education in connection with the Court's activities when the source is indicated and the reproduction is free of charge[15]. The same policy applies to documents retrieved from EUR-Lex whose documents are allowed to be reused in conjunction with the Commission Decision of 12 December 2011 on the reuse of Commission Documents[16] for commercial and non-commercial purposes given the source is

[12] `http://eur-lex.europa.eu/`
[13] `https://hudoc.echr.coe.int`
[14] `https://www.supremecourt.gov/`
[15] `https://echr.coe.int/Pages/home.aspx?p=disclaimer&c=`
[16] `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32011D0833`

Table 3 Corpus statistics

Corpus	# Doc.	# Tokens	Doc. Size (Avg. KB)	Doc. Size (Avg. Tokens)	Sentence length (Avg. Tokens)
ECHR	10	7,252	4	725	13
ECJ	10	53,044	32	5,304	32
USC	10	50,874	25	5,087	18
Total	30	111,170	20	3,705	21

Table 4 Statistics of corpora annotated with TimeML in literature.

Corpus	# Doc.	# Tokens	Doc. Size (Avg. Tokens)
TimeBank[20]	183	78,444 (61,000[21])	428.7
AQUAINT[22]	73	34,154	467.9
TempEval-3 Eval. (*Platinium*) (47)	20	~6,000[23]	~300
WikiWars (42)	22	119,468	5,430.4
Time4SMS (42)	1,000	20,176	20.2
Time4SCI (42)	50	19,194	383.9

acknowledged[17]. Documents published by US governmental institutions (such as the US Supreme Court) are in the public domain[18].

Legal documents often contain names of persons, especially court decisions. The documents in our corpus contain the names of the involved judges and the names of parties in a non-anonymized way. Names are considered personal data and need to respect the General Data Protection Regulation[19] (GDPR) which in the case of public data involves providing transparency with respect to the processing on request (Article 14 GDPR). Consent for the processing of personal data from the data subject is not required for public data.

For the purpose of temporal annotations we are mainly interested in the section of the court decisions describing the facts of a case, because we expect to find the most valuable temporal information about the chronology of a case in this section, whereas temporal information in other sections is expected to be relating to laws or previous cases. Therefore, we omitted in our corpus the “Legal framework” section of the documents from the ECJ.

The figures in Table 3 illustrate differences between documents depending on their source. Although we include documents from three different courts in this paper, the corpus statistics show that the documents in the ECJ and USC subcorpora are similar in terms of document size and length. The documents in the ECHR subcorpus are only one fifth in terms of size in comparison with the other two subcorpora. As stated previously, legal texts often make use of very long and complicated sentences to explain legal details, thus we also included the average sentence length in tokens for each corpus. We show that the sentences of the ECHR are roughly one third of length compared to the USC court decisions, and also tend to be shorter than the ones in the ECJ corpus. These numbers contrast with those relating to corpora from other domains and sources, such as Wikipedia articles (25.1 words per sentence (18)), the CONLL 2007 corpus of documents from the Wall Street Journal (24 and 23.4 tokens per sentence in training and test data, respectively (31)) and the basic corpus of everyday documents (33), including all kind of common texts, such as banking or

[17] https://eur-lex.europa.eu/content/legal-notice/legal-notice.html#droits

[18] https://www.copyright.gov/title17/92chap1.html#105

[19] Regulation (EU) 2016/679.

[20] http://www.timeml.org/timebank/documentation-1.2.html

[21] The website just mentions 61k non-punct tokens, the other figure was extracted from (42).

[22] http://www.timeml.org/timebank/aquaint-timeml/aquaint_timeml_1.0.tar.gz

[23] Just approximate figures were provided (47).

shopping documents (with an average of 17.2 words per sentence). Regarding the amount of documents in each corpus, Table 4 provides an overview (extracted from previous literature (47)) of the size of referential corpora manually annotated with TimeML. These figures provide evidence that despite the fact that we have less documents per corpus our corpus is substantially bigger in terms of tokens than most of the previous corpora.

4.2 Annotation

For each subcorpus (ECJ, ECHR and USC), the ten documents were selected at random. In order to compare the results of different temporal annotation tools, all thirty documents have been annotated in multiple steps. In the first part of the annotation process, two different annotators performed the manual annotation of the documents following the TimeML guidelines[24]. Once manual annotation, which was done independently by two persons using General Architecture for Text Engineering (GATE) (7), was completed, they met to create a gold standard with annotations agreed by both annotators. When doubts arose, the TimeML guidelines were consulted specifically looking for similar cases; if the doubt persisted, also the TIDES TIMEX2 guidelines[25] were examined, as referred to in the TimeML annotation guidelines. However, due to the particularities of the legal domain, some annotation decisions needed further discussion as shown in the following examples:

1. The word `now` is heavily used in legal documents and was only annotated when it was not used as an adverb, hence the meaning is not `currently` or `at the moment`. For instance in the case ECJ C-457/12, *[...] so the provision is* `now` *worded as follows [...].*

2. For the annotation of references to the present time, some taggers use the *PRESENT_REF* token as a value, while others normalize to a date (usually the creation date). We decided for the legal domain we should follow the latter approach, since all the documents in the corpus contain this information and humans would also be able to derive it.

3. Legal documents, especially judgments, often contain references to previous court decisions in the legal grounding of a decision. The citation of such preceding cases depends on how decisions of such courts are usually referenced. Typically, a year is contained in the citation and annotated as a temporal reference. Temporal information contained in identifiers used to refer to collections of court decisions (e.g. *2006*I) or included in the document identifier, should not be annotated (e.g. EC:C:*2013*:180).

4. Expressions such as `the date indicated`, appearing for instance in the excerpt "*the application lodged on **the date indicated** in (...)*" are not considered as temporal references but as co-references, being therefore not annotated in the gold standard, since a temporal tagger would not be expected to do so.

The discussion between the two annotators resulted in the creation of two gold standards *StandardTimeML* and *LegalTimeML*:

1. **StandardTimeML** annotates all the TEs following the TimeML guidelines, and uses the *PRESENT_REF*, *PAST_REF* and *FUTURE_REF* tokens as usually done in the domain.

2. **LegalTimeML** annotates just the TEs relevant to the narratives of the judgment, following the particularities in the legal domain previously discussed (no dates in legal references, normalize to dates...). As per the *StandardTimeML* annotation set, it follows the guidelines but does not annotate all the expressions, being therefore a subset considering domain particularities.

[24] `https://catalog.ldc.upenn.edu/docs/LDC2006T08/timeml\_annguide_1.2.1.pdf`
[25] `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-timex2-guidelines-v0.1.pdf`

The Inter-Annotator Agreement (IAA) between both gold standards is high (0.95), as well as Cohen's kappa (6) (0.94) and Scott's Pi (40) (0.94), indicating that the normalization of the TE's that are included in both annotation sets have a high agreement. If we check differences between annotations, we find there are an average of 13.1 common TEs per document, 0.3 partial coincidences and about 16.2 TEs that are not in the *LegalTimeML* but appear in the *StandardTimeML*. The recall among both annotation sets is of 0.44 while precision is of 0.90, which confirms that a lot of TEs are not relevant for the case timeline (44% with regard to the ones annotated following the full TimeML standard), but that the way to tag them by the annotators is highly similar.

4.3 Tagging

Once the corpus was collected, the following temporal taggers: HeidelTime (43), SUTime (4) GUTime (which is part of the TARSQI toolkit) (48), CAEVO (3), ClearTK-TimeML (1), SYNTime (50), TERNIP (32), TIPSem (22), USFD2 (9) and UWTime (21) were executed over our legal corpus, as they represent the different approaches available and are the most widely used in literature. These temporal taggers will be introduced in Section 5. HeidelTime was used in its configuration for narrative text. GUTime was used as a part of the TARSQI toolkit, using it alone with the preprocessor in the pipeline. Since the code available online was just able to annotate an specific corpus, USFD2 was slightly modified in order to annotate any input and to generate TIMEX3 tags as output[26]. All other taggers were used with default parametrization.

The output of the taggers which generated offline annotations (such as GUTIME/TARSQI) were modified in order to be comparable with the output of the rest of the taggers and ensure they were readable by GATE. These processes were executed using a new coded converter, which added the temporal annotations to the document and excluded non-temporal entities. Once the outputs of all the taggers were in the same format, they were loaded into the same GATE document, which contained twelve annotation sets (two for the manually-created gold standards and one for each of the ten temporal taggers).

4.4 Final Corpus

The final documents have been generated in several formats.[27] First, as GATE XML documents, that facilitate the storage of different annotation sets and also the visual and numerical comparison of the different sets. Second, a set of TimeML documents (TML) is provided for each of the manual gold standards. These documents contain the same annotations as in the correspondent annotation set in the corresponding GATE document, but makes the comparison with the output of other temporal taggers easier, as it is in the *official* TimeML format. Also a set of TML documents without any tag is provided to facilitate testing. These TML documents have been validated using the TimeML validator[28] from TempEval-3[29], so it is guaranteed that they fulfill the guidelines of the TimeML standard. Finally, all original documents are stored as TXT-files; these documents are of similar size in terms of kilobyte and length in tokens as shown in Table 3.

5 Temporal Taggers

Many of the temporal taggers described in the literature over the last few years are no longer available, not maintained, or just work for previous annotation schemas like the formerly mentioned TIMEX2. Some examples are DANTE (27), TEA (14), JU_CSE (19) or ManTIME (11). Therefore, we focus on the most widely used active temporal taggers which are often cited in literature and report good results on corpora from different domains, or have successfully participated in well-known temporal challenges, such as TempEval-3[30].

Table 5 provides an overview of the temporal taggers under investigation for which an implementation is freely available. The first column is used to refer to particular temporal taggers later on.

[26] The functionality and the rules were not modified.

[27] The final corpus can be downloaded at: `https://tempcourt.github.io/TempCourt/`

[28] `http://www.cs.york.ac.uk/semeval-2013/task1/data/uploads/timeml-validator-1.1a.tar.gz`

[29] `https://www.cs.york.ac.uk/semeval-2013/task1/`

[30] `https://www.cs.york.ac.uk/semeval-2013/task1/`

Table 5 Overview of temporal taggers. (*) Not all the types are covered.

Temporal Tagger	Approach	Identification	Normalization	Events	Relations
HeidelTime (HE)	rule-based	✓	✓	-	-
SUTime (SU)	rule-based	✓	✓	-	-
GUTime (GU)	hybrid	✓	✓	✓	✓
CAEVO (CA)	hybrid	✓	✓	✓	✓
ClearTK (CL)	machine-learning	✓	-	✓	✓
SynTime (SY)	rule-based	✓	-	-	-
TERNIP (TE)	rule-based	✓	✓	-	-
TIPSem (TI)	hybrid	✓	✓	✓	✓
USFD2 (US)	hybrid	*	*	-	*
UWTime (UW)	hybrid	✓	✓	-	-

The following aspects will be discussed for each tagger: supported languages, used approach, covered functionality, parametrization options, implementation language, availability, integration and interoperability with other software and dependencies on other resources and required installations.

5.1 Tasks of Temporal Taggers

The functionalities of temporal taggers can be classified into four categories as shown in Table 5. Some temporal taggers support all functionalities while other taggers require some additional tools.

- *Identification* means that the system is actually able to identify temporal expressions in a text compared to other systems which are only used for normalization of already tagged texts.
- *Normalization* refers to the ability to represent temporal information in the written text into the corresponding standard value following the ISO 8601 norm, which can be further processed. For instance expressions like ‘the next day’ refer to the day before which might be indicated with an explicit date in the text, and the temporal tagger is able to normalize this expression and assign the actual date as the value to the temporal annotation.
- *Events* are real-world situations at a particular time and are classified into seven categories, such as `OCCURENCE, STATE` or `REPORTING`, in the TimeML standard (38).
- *Relations* indicate a certain connection between events, times or a mixture of both usually classified into temporal `TLINK`, subordination `SLINK` and aspectual `ALINK` links (38).

5.2 Approaches

The detection of temporal expressions in a text is based on different approaches. Some taggers use rules for both identification and normalization tasks, while others use Machine Learning for the former. Also hybrid approaches have been proposed in literature. Nevertheless, it must be noted that normalization is generally tackled using rules, even when the identification is done otherwise.

5.2.1 Rule-based Approach

Temporal information is detected based on manually created rules (e.g. regular expressions), which need to cover all possible variations of how temporal information might be expressed. Thoroughly created rules are expected to perform better than other approaches, but come with the disadvantage of being inflexible. A missing or erroneous rule will prevent the temporal tagger from finding a temporal expression.

HeidelTime (43) is a rule-based domain-sensitive temporal tagger. Available for more than 200 languages (just 13 of them based on manually developed resources, the rest of them being automatically created), it offers the option to select from four different text categories: *News, Narratives, Colloquial* and *Scientific*, the last two are only available for English. HeidelTime covers both TE identification and normalization, having different strategies for each domain. HeidelTime, implemented in Java, can be used as a standalone version[31], or integrated in other pipeline environments like the General Architecture for Text Engineering (GATE) (7) or a UIMA[32] pipeline. In spite of being one of the most popular temporal tagging tools, to the best of our knowledge, it has never been used in the legal domain.

SUTime (4) is the Stanford CoreNLP (26) annotator for temporal expressions. SUTime is a rule-based temporal tagger built on the TokenRegex tool (5) (a pattern definition service also part of CoreNLP), able to both identify and normalize TEs. SUTime produces TimeML/TIMEX3 tags with new attributes not included in the standard, such an alternative value more flexible than the one covered by the standard. SUTime presents several related limitations (as analyzed by the authors themselves in (4)) and offers no domain adaptation. SUTime is available as part of the CoreNLP pipeline as a Named Entity Recognition (NER) system for different languages. Still, the tool works better in English than in other languages. The Java code[33] is available online, and also a GATE plugin and a Python wrapper have been developed[34].

SynTime (50) is a rule-based tagger that proposes a *type-based* approach as it defines different types of tokens (*time tokens, modifiers* and *numerals*) with similar syntactic behaviour and builds heuristic rules on these types instead of doing it on strings or regular expressions. As the types are domain independent and the rules work on types, the system is designed to be domain and language independent; nevertheless, to work in different domains or languages, more tokens need to be added for each type. SynTime only performs TEs recognition, and does not normalize them. For initialization, both tokens and regular expressions over them are collected for the independent temporal tagger SUTime (4). It is written in Java and available online[35]. It uses the Stanford CoreNLP library for Part of Speech (POS) disambiguation.

TERNIP (Temporal Expression Recognition and Normalisation in Python) (32) is a rule-based Python 2.7 library that identifies and normalizes TEs. The rules used for both subtasks can be easily extended. It only covers English and provides no domain particularities. It can be used as an API or be integrated as a GATE processing resource, via an XGAPP file (a GATE application file format) available with the code[36]. TERNIP relies on the Natural Language Toolkit library (NLTK) (24).

5.2.2 Machine-learning-based Approach

In contrast to rule-based approaches machine-learning based temporal taggers do not rely on previously created rules to identify temporal expressions. Using machine-learning techniques makes temporal taggers much more flexible and enables them to detect temporal expressions in an unexpected form, however it requires a good pre-trained model based on a large annotated corpus that supports a variety of temporal expressions which can be expected later in the document to be tagged with temporal expressions. A poor training set with missing variations of temporal expressions will result in a poor performance of the temporal tagger in terms of *precision*[37] and *recall*[38].

ClearTK-TimeML (1) is a system that identifies temporal information in English texts using external machine-learning tools. It uses specific annotators modelled as a BIO[39] token-chunking (for extent/identification of the expressions) or as a multiclass classification task (for types and attribute classification). The TIMEN normalisation tool (23) is suggested for the normalization task as this is not covered by ClearTK-TimeML. The features used are the ones proved to be the most successful in previous independent temporal taggers, and are extracted by a morpho-syntactic annotation pipeline with tools like OpenNLP and Apache. While ClearTK-TimeML does not offer domain-specific adaptions, the pipeline and the parameters can be customized by the user. It is written in Java and can be found online[40].

[31] https://github.com/HeidelTime/heideltime/

[32] https://uima.apache.org

[33] https://github.com/stanfordnlp/CoreNLP/tree/master/src/edu/stanford/nlp/time

[34] https://nlp.stanford.edu/software/sutime.shtml\#Extensions

[35] https://github.com/xszhong/syntime

[36] https://github.com/cnorthwood/ternip

[37] Fraction of the results identified which were correct.

[38] Fraction of the results that should have been found which were correctly identified.

[39] Beginning of, Inside of, Outside of a time expression.

[40] https://cleartk.github.io/cleartk/docs/module/cleartk_timeml.html

5.2.3 Hybrid Approach

Hybrid approaches combine rules with machine-learning. For instance creating rules of large corpus with machine-learning techniques to be manually refined afterwards.

GUTime (25) was developed at the Georgetown University originally for the temporal annotation of news. GUTime was subsequently incorporated into **TARSQI**, a modular system for automatic temporal annotation (48). The approach of GUTime is different from the temporal taggers previously mentioned, as it does not only use rules to find temporal expressions, but it also applies a hybrid approach of rules and machine-learning techniques. The hand-crafted rules serve in GUTime as a basis for temporal annotations that are extended by additional machine-learning ones discovered using the C4.5 algorithm (36), i.e. rules to support term disambiguation. The TARSQI framework is also able to extract events and relations from English texts. TARSQI is written in Python[41] and well described[42].

CAEVO (3) (CAscading EVent Ordering) is a sieve-based architecture, which uses twelve different classifiers (both rule-based and machine-learning), pipelined in a cascade way, starting with the one with the highest precision. Even when these classifiers work individually, some transitivity constraints are imposed; also the order of the classifiers can be modified, and new sieves can be added. In contrast to other taggers, CAEVO focuses on the extraction of temporal relations for event ordering, producing *dense* temporal graphs where events and temporal expressions are heavily connected. CAEVO is an expansion of NavyTime (2) and reuses part of the code of ClearTK-TimeML (1) for part of its sieves. It works just for English texts and has no domain adaptations. It is written in Java, and it is available online[43].

TIPSem (22) (Temporal Information Processing based on Semantic information) is an hybrid temporal tagger able to extract temporal information from English and Spanish. It uses both Semantic Role Labeling (13) and Conditional Random Field (CRF) (20) models. Different features are used by CRF recognition models, such as morphological or syntactic considerations at token level, along with semantic level ones such as the Role, the Governing Verb or Lexical Semantic information for each token. Similar features are used at tag level for classification. Finally, the relation extraction features differ depending on the type of relation. TIPSem tackles therefore all the temporal tasks. The Java code is available online[44], but it requires installation of additional software, and also optional libraries for certain languages (such as Spanish).

USFD2 (9) is a temporal tagger focusing on TEs and relations, using a rule-based approach for TEs and both rules and the NLTK's Maximum Entropy classifier for relations. USFD2 obtains a good recall with a smaller set of rules when compared with other taggers, since they consider specific heuristics for scpecific tags, such as `DATE`s and `DURATIONS`s as Temporal Expression types, that are the most common. It only works for English. The Python code of USFD2 is available online[45], but it must be noted that it is developed for the evaluation of specific datasets, so it must be slightly modified for custom use. This has been done for the results on our corpus described in this paper.

UWTime (21) follows a hybrid approach, using a Combinatory Categorial Grammar (CCG) (41) parser with hand-crafted rules and learning. UWTime just tackles the recognition and normalization of temporal expressions. It uses features such as surrounding tokens and POS, lexical and dependency information, and relies on techniques such as AdaBoost (12) for optimization. UWTime is only available in English with no domain particularities. It can be downloaded online[46], used as an API or as a server. UWTime relies on Stanford CoreNLP software.

6 Evaluation and Results

The final step of our research methodology involved a comparison of the effectiveness of all ten taggers on the two gold standards, along with the analysis of the results.

[41] `https://github.com/tarsqi/ttk`
[42] `http://timeml.org/tarsqi/index.html`
[43] `https://github.com/nchambers/caevo`
[44] `https://github.com/hllorens/otip`
[45] `https://github.com/leondz/usfd2`, `https://code.google.com/archive/p/usfd2/`
[46] `https://bitbucket.org/kentonl/uwtime-standalone`

6.1 Evaluation Methodology

After having all documents annotated with the ten different temporal taggers we evaluated the results, for which we used the typical *precision*, *recall* and *F-measure* metrics, which are commonly used in literature for the evaluation of extraction and normalization of temporal annotations (43). Precision is defined as the share of correctly identified items in percent compared to all identified items; whereas recall is defined as the number of items correctly found compared to the number that should have been found. The third measure we included in the evaluation, the *F-measure*, describes a weighted average between precision and recall (37). It is worth noting that we elected to provide both the *strict*-F-measure (which only considers completely correct and ignores partially correct annotations) and the *lenient*-F-measure, that admits partial annotations. The reason to do so is that while it is important to identify the complete temporal expression, it is also true that some taggers normalize correctly an expression even if they do not fully cover it. It also must be taken into account that in some cases the correct extent of a temporal expression is not clearly derivable from the guidelines, for this reason we decided that providing both measures would allow for the evaluation of both the degree of support with respect to the guidelines and the actual detection capabilities.

The evaluation process was designed in a way to avoid a bias or preference towards a particular temporal tagger. Therefore, the results of all taggers are consolidated in a single document with individual annotation sets for each tagger containing the temporal annotations and respective features. Each evaluation involves a key set (the correct reference) and a response set (the annotations to evaluate). Since the goal is to create gold standards for the legal domain, the manually annotated temporal expressions in both annotation sets, *LegalTimeML* (LTML) and *StandardTimeML* (STML), serve as the key sets. The annotation sets of each tagger act as the response set for each evaluation run. We therefore evaluated each automatic tagger for all three sections of the corpus (i.e. the documents from the three different legal sources) against each of the manually created gold standards *LegalTimeML* and *StandardTimeML* and calculated the *lenient* and *strict* precision, recall and F-measure.

All the temporal taggers were applied to the corpus with the standard configuration and there were no domain-specific modifications to achieve better results specifically for the legal domain[47]. The standard configuration was chosen so as to evaluate the out-of-the-box performance of each annotator and the suitability when applied to the legal domain. The average number of annotations per corpus in both Gold Standards (STML and LTML) and the various taggers are shown in Table 6, which illustrates the occurrences of different TIMEX3 annotation types (`DATE`, `DURATION`, `TIME`, `SET`) for each analysed corpus. It is clearly shown that the most used annotation type in court decisions is DATE. This result is not surprising as the date is considered to be sufficient in most cases as the actual time of the day is not relevant. Furthermore, deadlines in the legal domain usually indicate the end of the day and it is not important if an action is taken in the morning or in the afternoon. It must also be noted that the pattern of appearances of each of the TIMEX3 types does not fit any of those of the domains analyzed by Strötgen et al. (43) (news, narratives, colloquial and scientific).

Table 7 clearly shows that most taggers perform well on the short ECHR subcorpus and tend to find the same number of annotations as in the gold standard, especially if we focus on the *lenient* figures, showing that the errors are mostly in the extension of the tagging more than in its identification. In the ECJ and USC subcorpora (Tables 8 and 9 respectively) the number of annotations by the taggers differs from the gold standards, especially HeidelTime draws attention to its annotations in the ECJ corpus. When looking into the documents, the reason for this significant difference becomes obvious. The designators of European legal acts such as regulations and directives follow a special scheme which also includes the year when the legal act has been agreed. A typical designator of an EU directive is therefore, for instance ***2016***/*679*, which is considered to be a designator of a legal act but it is not a valuable temporal reference within a court decision.

[47] Except USFD2.

Table 6 Average number of annotation types per document for each corpus (***D****ate*,***Dur****ation*,***S****et*,***T****ime*).

	ECHR				ECJ				USC			
Tagger	**D**	**Dur**	**S**	**T**	**D**	**Dur**	**S**	**T**	**D**	**Dur**	**S**	**T**
StandardTimeML	11.6	1.3	1	0	31.5	4.3	2	2.7	35.7	5.6	3.5	4
LegalTimeML	10.1	1.3	1	0	16.8	4.3	1.5	3	9.1	5.4	1.5	0
HeidelTime	11.4	1.7	1	0	68.1	5.3	1	1	41.6	5.6	1.5	2
SUTime	11.3	2	0	0	39.1	3.9	1.3	1.3	46.9	7.9	1.5	2.7
GUTime	11.7	0	0	0	31.4	1	0	0	37.3	2	0	0
CAEVO	11.1	1.8	0	0	36.7	5.8	1	1.5	39.9	9.4	1.5	3
ClearTK	10.2	1	0	0	38.6	3.4	0	0	36.1	5.1	1	2
Syntime	11.5	0	0	0	39.1	0	0	0	47.8	0	0	0
TERNIP	11.7	1.7	0	0	30.3	3.6	0	0	33.3	5.6	1	0
TIPSem	13	1	0	0	38.4	2.6	0	0	-	-	-	-
USFD2	13.9	2	0	0	66.6	3.3	0	0	28.4	3.8	0	0
UWTime	11	2.5	0	0	-	-	-	-	-	-	-	-

Table 7 Evaluation results for the ECHR corpus for each temporal tagger, both for identification (two first columns, *lenient* and *strict*) and normalization (two last columns, *lenient* and *strict*). The first row (in white) corresponds to results against the *StandardTimeML* gold standard, while the second (in gray) corresponds to the *LegalTimeML* gold standard.

	lenient			strict			lenient+ value			strict+ value		
A	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
HE	**0.99**	**0.99**	**0.99**	0.84	0.84	0.84	0.78	0.78	0.78	0.78	0.78	0.78
	0.88	**0.99**	**0.93**	0.71	0.80	0.75	0.67	0.75	0.71	0.64	0.72	0.68
SU	0.88	0.87	0.88	0.85	0.84	0.84	0.78	0.78	0.78	0.76	0.75	0.75
	0.76	0.85	0.80	0.71	0.80	0.76	0.66	0.74	0.79	0.64	0.72	0.68
GU	0.96	0.93	0.94	**0.95**	0.92	**0.93**	**0.86**	0.84	0.85	**0.86**	0.84	**0.85**
	0.84	0.92	0.88	**0.83**	**0.92**	**0.87**	0.74	0.82	0.78	**0.74**	0.82	**0.78**
CA	0.88	0.87	0.87	0.83	0.82	0.82	0.78	0.78	0.78	0.75	0.75	0.75
	0.75	0.85	0.80	0.70	0.79	0.74	0.65	0.74	0.69	0.64	0.72	0.67
CL	0.92	0.78	0.85	0.34	0.32	0.35	-	-	-	-	-	-
	0.80	0.77	0.78	0.33	0.32	0.33	-	-	-	-	-	-
SY	0.98	0.93	0.96	0.83	0.79	0.81	0	0	0	0	0	0
	0.86	0.93	0.90	0.70	0.76	0.73	0	0	0	0	0	0
TE	0.94	0.95	0.95	0.92	**0.93**	0.92	**0.86**	**0.88**	**0.87**	0.85	**0.86**	**0.85**
	0.83	0.95	0.89	0.80	**0.92**	0.85	**0.75**	**0.86**	**0.80**	0.72	**0.83**	0.77
TI	0.78	0.85	0.81	0.64	0.70	0.67	0.64	0.71	0.67	0.63	0.69	0.66
	0.69	0.86	0.76	0.62	0.77	0.69	0.64	0.79	0.71	0.61	0.76	0.68
US	0.73	0.61	0.67	0.69	0.58	0.63	0	0	0	0	0	0
	0.65	0.62	0.64	0.61	0.58	0.60	0	0	0	0	0	0
UW	0.90	0.53	0.67	0.51	0.30	0.38	0.55	0.33	0.41	0.51	0.30	0.38
	0.86	0.58	0.69	0.48	0.32	0.38	0.51	0.34	0.41	0.48	0.32	0.38

6.2 Results

From the results shown in Tables 7 (ECHR), 8 (ECJ) and 9 (USC), we can see that the performance of the individual temporal taggers is quite similar for each section of the corpus. Furthermore, the numbers for all three measures that have been calculated are unexpectedly high for some taggers in comparison to the application of temporal taggers (out of the box without any domain-specific modifications) in the case of non legal text. They tend to be nevertheless less performant than results previously reported by taggers in general evaluations[48] (4).

[48] `https://github.com/HeidelTime/heideltime/wiki/Evaluation-Results`

Table 8 Evaluation results for the ECJ corpus for each temporal tagger, both for identification (two first columns, *lenient* and *strict*) and normalization (two last columns, *lenient* and *strict*). The first row (in white) corresponds to results against the *StandardTimeML* gold standard, while the second (in gray) corresponds to the *LegalTimeML* gold standard.

	lenient			**strict**			**lenient+ value**			**strict+ value**		
A	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
HE	0.48	0.95	0.64	0.47	0.94	0.63	0.47	**0.94**	0.62	0.47	**0.93**	0.62
	0.27	0.97	0.42	0.26	**0.96**	0.41	0.26	**0.94**	0.40	0.26	**0.93**	0.40
SU	0.81	0.97	0.88	0.79	**0.95**	0.86	0.78	0.93	0.85	0.77	0.92	0.84
	0.44	0.95	0.60	0.43	0.93	0.58	0.41	0.90	0.57	0.41	0.89	0.56
GU	**0.97**	0.87	0.91	**0.97**	0.86	**0.91**	0.94	0.84	0.89	0.94	0.84	0.88
	0.51	0.82	0.63	0.50	0.82	0.62	0.48	0.78	0.60	0.48	0.78	0.60
CA	0.89	0.74	0.81	0.85	0.70	0.77	0.86	0.71	0.77	0.85	0.70	0.77
	0.49	0.74	0.59	0.46	0.70	0.56	0.46	0.70	0.56	0.46	0.69	0.55
CL	0.77	0.88	0.82	0.32	0.36	0.34	-	-	-	-	-	-
	0.42	0.88	0.57	0.18	0.37	0.24	-	-	-	-	-	-
SY	0.89	**0.99**	**0.93**	0.81	0.90	0.85	0	0	0	0	0	0
	0.49	**0.98**	0.65	0.46	0.92	0.61	0	0	0	0	0	0
TE	**0.97**	0.88	0.92	0.96	0.88	**0.91**	**0.96**	0.87	**0.91**	**0.95**	0.87	**0.91**
	0.54	0.89	**0.67**	**0.53**	0.88	**0.66**	**0.53**	0.88	**0.65**	**0.52**	0.87	**0.65**
TI	0.72	0.81	0.76	0.64	0.72	0.68	0.62	0.70	0.65	0.61	0.69	0.65
	0.41	0.83	0.54	0.37	0.75	0.49	0.35	0.71	0.47	0.34	0.70	0.46
US	0.31	0.54	0.39	0.29	0.51	0.37	0.02	0.04	0.03	0.02	0.03	0.02
	0.20	0.65	0.31	0.19	0.61	0.29	0.02	0.06	0.03	0.02	0.05	0.02
UW	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-

On the ECHR corpus most taggers perform equally well when *strictly* evaluated, while GUTime provides the best results, closely followed by TERNIP. On the contrary, TIPSem, USFD2 and UWTime are not as performant. This is because the ECHR uses fully qualified dates (e.g. 10 January 2017) and does not include many references to other court decisions. These results fall when we look at the normalization values. It also must be noted that most taggers (except of GUTime, SynTime and TERNIP) struggle with identifying dates denoting the birthdates of the persons involved in the cases and case numbers, with some also normalizing them. It must be noted how big differences between *lenient* and *strict* values such as those of UWTime or ClearTK-TimeML do not always affect in terms of differing in the extent of the tag, but it also impacts in the normalization values. For instance, if instead of marking up 'October 13', just 'October' is marked, the *lenient* score will count it as positive, the *strict* will not, but the normalization will for sure be wrong.

In the ECJ corpus one outlier in the figures can be spotted immediately, which is the precision of the HeidelTime annotations that is significantly different from its other precision values across each section of the corpus. The much better performance of GUTime in the ECJ corpus can be explained by the fact that it does not annotate numbers referring to collections of judgments (such as TIPSem or ClearTK-TimeML do).

The USC corpus is slightly different to ECHR and ECJ as it uses a different date format and it also repeats part of the text in the judgment, which leads to poorer performance as incorrect annotations are also repeated.

Different date formats are a typical challenge which occur when applying temporal taggers to a corpus. Typically dates found across all evaluated documents are fully qualified dates containing a day, the month in full and a year. The format in which these dates are provided are different for European and American sources of legal documents. The date in Europe is usually indicated in the format Day, Month, Year (e.g. 10 January 2017), whereas the American date format is "Month DD, YYYY" (e.g. January 10, 2017). This particular difference in the date format has been processed correctly by some taggers, such as HeidelTime

Table 9 Evaluation results for the USC corpus for each temporal tagger, both for identification (two first columns, *lenient* and *strict*) and normalization (two last columns, *lenient* and *strict*). The first row (in white) corresponds to results against the *StandardTimeML* gold standard, while the second (in gray) corresponds to the *LegalTimeML* gold standard.

	lenient			**strict**			**lenient+** **value**			**strict+** **value**		
A	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
HE	0.83	0.94	0.88	0.81	0.92	**0.86**	0.79	**0.90**	**0.84**	0.79	**0.89**	**0.83**
	0.29	0.97	0.44	0.26	0.88	0.40	0.20	0.67	0.31	0.19	0.64	0.29
SU	0.75	**0.99**	0.85	0.72	**0.95**	0.82	0.67	0.88	0.76	0.66	0.86	0.75
	0.25	**0.98**	0.40	0.23	**0.90**	0.36	0.18	0.72	0.29	0.17	**0.69**	0.28
GU	0.84	0.78	0.81	0.71	0.66	0.69	0.67	0.62	0.65	0.65	0.60	0.62
	0.25	0.69	0.36	0.16	0.45	0.23	0.12	0.34	0.18	0.10	0.27	0.14
CA	0.77	0.90	0.82	0.72	0.84	0.77	0.73	0.85	0.78	0.71	0.83	0.76
	0.23	0.82	0.36	0.21	0.72	0.32	0.21	**0.73**	0.33	0.20	**0.69**	0.30
CL	0.85	0.84	0.84	0.81	0.79	0.80	-	-	-	-	-	-
	0.30	0.89	0.45	0.26	0.78	0.39	-	-	-	-	-	-
SY	0.85	0.98	**0.91**	0.78	0.91	0.84	0	0	0	0	0	0
	0.28	**0.98**	0.44	0.24	0.84	0.37	0	0	0	0	0	0
TE	**0.93**	0.86	0.90	**0.90**	0.83	**0.86**	**0.86**	0.79	0.83	**0.85**	0.78	0.81
	0.32	0.90	**0.48**	**0.29**	0.81	**0.43**	**0.25**	0.69	**0.37**	**0.23**	0.64	**0.34**
TI	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-
US	0.50	0.30	0.37	0.39	0.23	0.29	0.08	0.02	0.08	0.02	0.01	0.02
	0.16	0.28	0.21	0.07	0.13	0.09	0.08	0.14	0.10	0.03	0.05	0.04
UW	-	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-

and SUTime, annotating both versions as a single date. GUTime however was not reliable in this context, despite the fact that it is the best tagger in the other corpora. It either detected only one part of the American-formatted date (e.g. January 10) or it treated both parts of the same date as two different annotations.

The performance of GUTime in terms of precision, recall and F-measure is pretty good over all three subcorpora. However, GUTime performs poorly on the USC corpus. Inspecting the GUTime annotations in this corpus confirms the fact that GUTime has a hard time recognizing dates in the American format, as already pointed out above, an issue that is also reflected in normalization figures (where TERNIP maintains the performance from the other subcorpora).

In summary, although the results of the evaluation are promising it is worth noting that legal documents, especially court decisions, have some particularities (such as those highlighted in Section 3) which cause some stumbling blocks for automatic temporal taggers being applied out-of-the-box. An example of this would be the case of *'dec.'*, a non-temporal expression that appears when citing *decisions on admissibility*[49] that most taggers (such as CAEVO or SUTime) normalize as *December*.

With regard to the comparison between the two reference standards, if we check the differences between figures and focus on the recall (since the taggers are not trained for the particularities of this annotation set, the precision is obviously not expected to be high and does not indicate the tagger's usefulness), we see that the best taggers remains more or less the same (GUTime, TERNIP, SUTime and HeidelTime, since although SynTime performs well in terms of recognition it does not provide a value).

6.3 Comparative Analysis of Several Temporal Taggers

The thorough analysis of the corpus documents and the manual inspection of the most frequent errors of the taggers led to the synthesis of a collection of test cases that present the phrases prone to cause errors.

[49] `http://www.echr.coe.int/Documents/Note_citation_ENG.pdf`

The most salient results are described below, where the output of the tagger is represented in bold and the correct tag is underlined.

HeidelTime is able to identify temporal modifiers (e.g. *at least five years*) automatically and add the feature to the annotation. However, it fails to detect the correct date format (e.g. DD/MM/YYYY vs MM/DD/YYYY) as well as failing to recognize the indication of the age of mentioned persons (e.g. "*a **62-year-old** woman*"). It does not normalize expressions like *today* and annotates them with the value `PRESENT_REF`. In legal texts, it tags as TEs references to other documents or IDs (e.g. *"No **1612/68**"*, *"No. 15-**1031**"*, *"See Pet. for Cert. **5-7**"*). It also has an interval option that does not work well in this kind of document.

SUTime also fails to identify the correct date representation form (e.g. DD/MM/YYYY vs MM/DD/YYYY). In addition, SUTime exhibits inconsistencies when parsing the same expression in different paragraphs, and it also wrongly annotates expressions like *"**fall**"*, *"**may**"* as temporal expressions although they refer to an action *"to fall"*, *"may"* instead of the season. SUTime also has some limitations with respect to ambiguity resolution or non-whole numbers recognition.

Although GUTime has a good performance in general, sometimes it does not normalize some expressions and has problems with some ways to represent hours (e.g., it does not recognize "*(...) between 12.15 and 18.45*", nor if it was expressed as "*12:15 and 18:45*", it just recognizes "***12h15** and **18h45***"). Also some DURATIONs are not recognized, series or dates neither (in "*15 and **16 December 2008***" it just recognizes the part in bold) , and sometimes it tags expressions that look like years, such as "*EUR **2000***".

CAEVO does normalize DATEs in the format DD/MM/YYYY as MM/DD/YYYY, so it does not even recognize the ones not fitting it, such as "*25/03/2016*". It also partially annotates expressions such as "*On the next **day***" (categorizing it as a DURATION) and tags separately "***once a week***", as a `PAST_REF DATE` and a `DURATION`, respectively. It also does not recognize 15 in "*15 and **16 December 2008***", and tags "***62-year-old** woman*", year-like expressions as *"**§1101**"* and time-like expressions as "*Order in No. **2:10**-cv-02698 (WD Tenn.)*". Finally, it also tags separately "*sentenced to **a year** and **a day** in prison*".

Similarly to GUTime, ClearTK-TimeML does not recognize TIMEs when expressed as in "(...) between 12.15 and 18.45"; it does not either recognize expressions like "09/01/1981" as DATEs. Some DURATIONs are also not recognized (e.g. "at least five years"), and tags expressions such as "**May**" or "**62-year-old** woman". It just annotates partially expressions such as "23 **January 2013**" or "once **a week**" (that is categorized as a DURATION).

SynTime just normalizes to the date when it is executed. Although it is able to recognize expressions such as "***15** and **16 December 2008***", it fails when it finds expressions such as "*as amended by Council Regulation (EC) No **1791/2006 of 20 November 2006***", where it annotates all in bold, not just the underlined correct part. It also seems to recognize all four-digit expressions as years (e.g.. "*See 10 U. S. C. §**1408**(c)(1).*", "*So. 3d **1264, 1269-1272***") and ambiguous expressions as "***may***", "***the second***" or "***fall***", but fails to fully annotate some temporal expressions (e.g. "*per **month***", "*May 15, **2017***").

TERNIP tags expressions such as "*EUR **2000***", "***may***", ”***fall***", but fails to identify some DATEs and DURATIONs. It also does not identify 13 in "*13 and 27 October 2008*", but is on the other hand is able to recognize misspelled temporal expressions such as "***eighth months***" (even if it is not correctly normalized). It also tags "*303, 98 Stat. **2045**, 21 U. S. C. §853(a)(1),*" as DATEs expressions .

TIPSem is not able to annotate some of the documents in the corpus (namely the ones from the USC subset), and does not recognize the first DATE in the ECJ subset, expressed as in the format `DD Month YYYY`; since it recognizes in the rest of the document without a problem, it is probably due to a lack of a syntactic/semantic context for it. It tags expressions such as "*Directives **2004/83, 2005/85** and 2003/9]*" or "***Article 5 of Directive 2008/115***", "**Directive 2001/42**" or "***the judgment of 28 February 2012***". It also tags expressions such as "***MON 810***" or random numbers or words as "***4,285***", "*(in **euros)i***", that tends to mark as `FUTURE_REF`. On the other hand, it does not recognize some dates, as "*29/02/2016*", but it does so with a similar one like "***28/09/2016***".

USFD2 is unable to parse some of the documents in the corpus, throwing errors when trying to normalize expressions it considers out of the range and warnings for some ASCII codes. It also tags some numbers randomly, such as in "*amending Regulation (EEC) No **1612/68** and repealing Directives*

*64/**221**/EEC, 68/**360**/EEC*" and always normalizes DATEs to the present day. It does not recognize straightforward dates and tags ambiguous words even when they are a part of another word, such as in "*Soto**mayor***"; TIME expressions are categorized as DATEs.

Finally, UWTime is not able to parse long legal sentences, throwing several errors because of the lack of head rules defined for some of the expressions it finds. In our corpus, it was not able to annotate even a third of the documents.

The most commonly occurring errors in which the taggers fall, whether because they happen frequently in the text or because several taggers incur in them, are the following:

- Separation of whole SET expressions as "*Once a week*" into "*Once*" and "*a week*", converting one SET into a PAST_REF DATE and a DURATION.
- Not recognizing series of DATEs such as "*15 and 16 December*", but detecting the last DATE of such a series only.
- Separation of DURATIONs such as "*One year and one day*" into two different DURATIONs.
- In some documents (as also happens in other kinds of legal texts, such as in the previously mentioned transactional ones), some information is put into brackets, such as in "*before the expiry of a period of **[48] hours***"; usually generic temporal taggers are not able to detect them (for instance tagging in this concrete example just "*hours*").
- Tagging general ambiguous expressions such as "*fall*" or "*may*" or specific ambiguous ones such as the previously described case of "*dec.*".
- Tagging year-like expressions such as "*No **1612**/68*" or "*§**1408***"; most taggers tag every four-digit number as a year.
- Problems with dates expressed in the format "DD/MM/YYYY", frequently in identification but in some cases also in normalization.
- Identification of a currency as a year ("*EUR **2000***").
- Tagging of expressions such as "*62-year-old*".
- Most taggers do not take modifiers (mod) into account, probably because of the low ratio of appearance of SETs in other domains, despite the fact that they are extremely important in legal documents. Namely, HeidelTime correctly tagged[50] 17 out of 28 modifiers, while TERNIP tagged 10 out of 28. The remaining taggers tagged no modifiers (Fexcept of UWTime, in one of the few documents it tagged, but not correctly).
- The case of the quant and freq attributes is similar for SETs. While HeidelTime marks correctly 2 out of 11 quant, and marks incorrectly two freq as 1 (when it should be 1X), TERNIP just marks one quant (and incorrectly, since it must be in capital letters) out of 11 and no freq.

7 Conclusion

In this paper we pointed out the importance of temporal information contained in legal documents. An extensive state of the art analysis showed that the extraction of temporal information has been investigated for other domains but not for the legal domain.

Considering the specific requirements of temporal annotation in the legal domain, we identified a lack of corpora that can be used for the evaluation of temporal entity extractors. In order to fill this gap, we created a corpus of 30 documents from the European Court of Human Rights, the European Court of Justice and the United States Supreme Court, containing manually annotated temporal expressions. The

[50] Some cases, such as distinctions between *EQUAL_OR_LESS* / *LESS_THAN* (for UWTime) and *LATE* / *END* and *EARLY* / *START* (for TERNIP) were counted as errors.

corpus is presented in two forms: (i) a generic gold standard called *StandardTimeML*; and (ii) a domain-focused gold standard called *LegalTimeML*. The latter was tailored specifically for temporal dimensions that are important for the entailed legal case, namely the *temporal dimension of the legal process* and the *temporal dimension of the case*.

We also preformed an in-depth analysis of several state-of-the-art temporal taggers and performed a comparative evaluation against our corpus. The results of our analysis on the *StandardTimeML* gold standard shows that the best temporal taggers are quite effective when it comes to finding all possible temporal expressions in legal text, however they fail when they encounter misleading references to legal documents. This can generally be attributed to the fact that courts tend to use a clear structured language and absolute date formats. It is not surprising that the performance of the cross domain temporal taggers on the *LegalTimeML* gold standard is much less impressive, highlighting the need for tools and guidelines that are specifically tailored to particularities of the legal domain.

The work presented herein is a prerequisite for future work which focuses on the automatic extraction of timelines from legal text. In this context, it will also be necessary to evaluate existing event extraction techniques, with respect to the particularities of the legal domain. The combination of temporal information and legal events could result in the creation of a temporal events taxonomy, that would help in a better understanding of legal processes. Additionally, based on our analysis and experience working both with temporal expressions and events, we aim to develop a set of guidelines, which will be of benefit for the legal informatics community. Besides the extension of this work towards event extraction and timeline creation, the legal domain is also very language dependent. Documents published in various countries and jurisdictions are typically written in the national language. Therefore, an interesting avenue for future research is to analyze the performance of existing temporal taggers over legal corpora that are written in languages other than English.

Acknowledgement

This work was partially funded by a Predoctoral grant from the Programa Propio de la Universidad Politécnica de Madrid, and from a grant from Consejo Social de la Universidad Politécnica de Madrid. This work was supported by the Republic of Austria's Federal Ministry for Digital and Economic Affairs and the Jubiläumsfonds der Stadt Wien

References

[1] S. Bethard. ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Proceedings of the Workshop SemEval 2013*, pages 10–14. ACL, June 2013.

[2] N. Chambers. Navytime: Event and time ordering from raw text. In *Proceedings of the Workshop SemEval 2013*, volume 2, pages 73–77, 2013.

[3] N. Chambers et al. Dense event ordering with a multi-pass architecture. *Transactions of the ACL*, 2: 273–284, 2014.

[4] A. X. Chang et al. Sutime: A library for recognizing and normalizing time expressions. In *Proceedings of LREC 2012*, 2012.

[5] A. X. Chang et al. TokensRegex: Defining cascaded regular expressions over tokens. Technical Report CSTR 2014-02, Department of Computer Science, Stanford University, 2014.

[6] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[7] H. Cunningham et al. Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. *PLOS Computational Biology*, 9(2):1–16, 02 2013.

[8] Dell'Orletta et al. The splet–2012 shared task on dependency parsing of legal texts. In *Semantic Processing of Legal Texts (SPLeT-2012) Workshop Programme*, page 42, 2012.

[9] L. Derczynski et al. Usfd2: Annotating temporal expresions and tlinks for tempeval-2. In *Proceedings of the Workshop SemEval*, pages 337–340. ACL, 2010.

[10] L. Ferro, L. Gerber, I. Mani, B. Sundheim, and G. Wilson. Tides 2005 standard for the annotation of temporal expressions. Technical report, Technical report, MITRE, 2005.

[11] M. Filannino and G. Nenadic. Temporal expression extraction with extensive feature type selection and a posteriori label adjustment. *Data & Knowledge Engineering*, 100:19 – 33, 2015. ISSN 0169-023X. doi: https://doi.org/10.1016/j.datak.2015.09.002. URL http://www.sciencedirect.com/science/article/pii/S0169023X15000725.

[12] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[13] D. Gildea and D. Jurafsky. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288, 2002. doi: 10.1162/089120102760275983. URL https://doi.org/10.1162/089120102760275983.

[14] B. Han, D. Gates, and L. Levin. From language to time: A temporal expression anchorer. In *Temporal Representation and Reasoning, 2006. TIME 2006. Thirteenth International Symposium on*, pages 196–203. IEEE, 2006.

[15] S. Hellmann et al. NIF: An ontology-based and linked-data-aware NLP Interchange Format. 2012.

[16] D. Isemann et al. *Temporal Dependence in Legal Documents*, pages 497–504. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-41278-3.

[17] H. Ji, T. Cassidy, Q. Li, and S. Tamang. Tackling representation, annotation and classification challenges for temporal knowledge base population. *Knowledge and Information Systems*, 41(3): 611–646, Dec 2014. ISSN 0219-3116. doi: 10.1007/s10115-013-0675-1. URL https://doi.org/10.1007/s10115-013-0675-1.

[18] T. Kajiwara et al. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016: Technical Papers*, pages 1147–1158, 2016.

[19] A. K. Kolya, A. Kundu, R. Gupta, A. Ekbal, and S. Bandyopadhyay. Ju_cse: A crf based approach to annotation of temporal expression, event 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)*, volume 2, pages 64–72, 2013.

[20] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In C. E. Brodley and A. P. Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann, 2001. ISBN 1-55860-778-1.

[21] K. Lee et al. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the ACL*, volume 1, pages 1437–1447, 2014.

[22] H. Llorens et al. Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the Workshop SemEval*, pages 284–291. ACL, 2010.

[23] H. Llorens et al. Timen: An open temporal expression normalisation resource. In *Proceedings of LREC 2012*, pages 3044–3051, 2012.

[24] E. Loper and S. Bird. NLTK: The Natural Language Toolkit. *CoRR*, cs.CL/0205028, 2002. URL http://arxiv.org/abs/cs.CL/0205028.

[25] I. Mani et al. Robust temporal processing of news. In *Proceedings of the 38th annual meeting on ACL*, pages 69–76. ACL, 2000.

[26] C. D. Manning et al. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the ACL 2014, System Demonstrations*, pages 55–60, 2014.

[27] P. Mazur and R. Dale. The dante temporal expression tagger. In Z. Vetulani and H. Uszkoreit, editors, *Human Language Technology. Challenges of the Information Society*, pages 245–257, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-04235-5.

[28] P. Mazur et al. Wikiwars: A new corpus for research on temporal expressions. In *Proceedings of EMNLP 2010*, pages 913–922. ACL, 2010.

[29] A.-L. Minard, M. Speranza, et al. Meantime, the newsreader multilingual event and time corpus. In *Proceedings of LREC 2016, European Language Resources Association*, 2016.

[30] V. Naik, G. Vanitha, and S. Inturi. Reasoning in legal text documents with extracted event information. *International Journal of Computer Applications*, 28:8—13, 08 2011.

[31] J. Nivre et al. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of EMNLP-CoNLL 2007*, 2007.

[32] C. Northwood. TERNIP: temporal expression recognition and normalisation in Python. Master's thesis, University of Sheffield, 2010.

[33] D. Pellow et al. An open corpus of everyday documents for simplification tasks. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 84–93, 2014.

[34] J. Pustejovsky et al. The Timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK, 2003.

[35] J. Pustejovsky et al. TimeML: Robust Specification of Event and Temporal Expressions in Text. In M. T. Maybury, editor, *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium*, pages 28–34. AAAI Press, 2003. ISBN 1-57735-184-3.

[36] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993. ISBN 1-55860-238-0.

[37] C. J. V. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.

[38] R. Saurí, J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, and J. Pustejovsky. TimeML annotation guidelines. *Version*, 1(1):31, 2006.

[39] F. Schilder. Event Extraction and Temporal Reasoning in Legal Documents. In *Annotating, Extracting and Reasoning about Time and Events, International Seminar, Dagstuhl Castle. Revised Papers*, pages 59–71, 2005.

[40] W. A. Scott. Reliability of content analysis: The case of nominal scale coding. *Public opinion quarterly*, pages 321–325, 1955.

[41] M. Steedman et al. *Combinatory Categorial Grammar*, chapter 5, pages 181–224. Wiley-Blackwell, 2011.

[42] J. Strötgen and M. Gertz. Domain-sensitive temporal tagging. *Synthesis Lectures on Human Language Technologies*, 9(3):1–151, 2016.

[43] J. Strötgen et al. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of LREC 2012*, volume 12, pages 3746–3753, 2012.

[44] W. Styler IV et al. Temporal annotation in the clinical domain. *Transactions of the Association of Computational Linguistics*, 2(1):143–154, 2014.

[45] J. Tabassum et al. Tweetime: A minimally supervised method for recognizing and normalizing time expressions in twitter. *arXiv preprint arXiv:1608.02904*, 2016.

[46] N. Uzzaman et al. Event and Temporal Expression Extraction from Raw Text: First Step Towards a Temporally Aware System. *International Journal of Semantic Computing*, 04(04):487–508, 2010.

[47] N. UzZaman et al. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Proceedings of the Workshop SemEval 2013*, pages 1–9, 2013.

[48] M. Verhagen et al. Automating Temporal Annotation with TARSQI. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, pages 81–84. ACL, 2005.

[49] C. S. Vlek et al. Representing and evaluating legal narratives with subscenarios in a bayesian network. In *OASIcs-OpenAccess Series in Informatics*, volume 32. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[50] X. Zhong et al. Time expression analysis and recognition using syntactic token types and general heuristic rules. In *Proceedings of the 55th Annual Meeting of the ACL*, volume 1, pages 420–429, 2017.