TIMEN: An Open Temporal Expression Normalisation Resource

Hector Llorens*, Leon Derczynski†, Robert Gaizauskas†, Estela Saquete*

*University of Alicante
03690, Spain
hllorens,stela@dlsi.ua.es

†University of Sheffield
S1 4DP, UK
leon,robertg@dcs.shef.ac.uk

Abstract

Temporal expressions are words or phrases that describe a point, duration or recurrence in time. Automatically annotating these expressions is a research goal of increasing interest. Recognising them can be achieved with supervised machine learning, but interpreting them accurately (normalisation) is a complex task requiring human knowledge. In this paper, we present TIMEN, a community-driven tool for temporal expression normalisation. TIMEN is derived from current best approaches and is an independent tool, enabling easy integration in existing systems. We argue that temporal expression normalisation can only be effectively performed with a large knowledge base and set of rules. Our solution is a framework and system with which to capture this knowledge for different languages. Using both existing and newly-annotated data, we present results showing competitive performance and invite the IE community to contribute to the resource in order to solve the temporal expression normalisation problem.

Keywords: Temporal Information Processing, TimeML, timex normalisation, ISO 8601, Open Resource

1. Introduction

This paper addresses temporal expression processing in natural language, which is framed in the field of information extraction (IE). We present an open, extensible and state-of-the-art temporal normalisation library TIMEN (`http://www.timen.org/`), that improves upon all other publicly available system performances.

A temporal expression (or **timex**) is a linguistic expression referring to a time, period, or recurring pattern in time (for example *"May 2012"*, *"next month"*, *"3 hours"*, *"weekly"*). Timex annotation involves the recognition of these expressions and then their interpretation, resulting in an annotation encoding a standardised representation of the timex's semantics, e.g. an ISO 8601 compliant specification of a calendrical time. This interpretation task is called **timex normalisation**.

The comprehension of temporal expressions is critical to accurate processing of discourse semantics. Achieving this has been a long-standing research task (Mani and Wilson, 2000; Verhagen et al., 2010). Aside from its intrinsic importance for discourse understanding, understanding temporal information is crucial for language processing applications including question answering (Saquete et al., 2009), text summarisation (Daniel et al., 2003), information retrieval (Alonso et al., 2007) and knowledge base population (Ji et al., 2011).

Timexes can be recognised using machine learning and has been achieved with relatively simple feature sets (Llorens et al., 2011). However, any practical approach to timex normalisation requires a hand-crafted rule set. The scale of annotated data and intelligent reasoning required to automatically infer, for example, the rules about the date of Easter Sunday or the associations between hemispheres and seasons from just text is too great to be practically feasible.

Previous approaches to timex normalisation have included their own custom rule sets and typically reach 60%-90% accuracy depending on evaluation dataset. Scant efforts have been made to build upon prior systems' performance. Instead, each system has incorporated a new, unique built-in rule set that captures the majority of timexes for a given training set that arise from a limited range of phrase types. This performance cap demonstrates the inherent disadvantages of rule-based approaches. Neither closed, integrated nor proprietary rule bases are equipped to normalise unseen timexes, or to grow in order to handle them. Further, much of the normalisation effort is inherently language-independent, working with structures such as calendars and simple temporal constructs such as month and day names. Separating the logic for dealing with this from language-specific requirements enables effective normalisation across languages. Finally, the limited amount of temporally annotated training data and its restriction to the newswire genre suggest that the real accuracy rate of existing systems will be lower than their evaluation indicates.

In light of this, our overall research question is: How can we reach 100% normalisation accuracy in timex interpretation?[1] Towards this goal, the questions we address in this paper are:

1. How can we create a high-performance multi-lingual timex normalisation system? E.g. how will we normalise timexes to improve upon prior work?
2. How can a normalisation system be made permanent, reusable and extensible? That is, what features must such system have so that it can grow over time?

[1] We accept that achieving maximum accuracy may be complex and time-consuming and that some unconquerable errors may be caused by poor-quality input documents. Nonetheless we are confident much higher normalisation accuracy can be achieved.

To address the first we propose an approach involving handcrafted rules and a rule processing engine. This builds on various rule sets extracted or intuited from previous systems and includes an evaluation component.
In response to the second, we propose a rule creation strategy that includes a constant performance evaluation component and is driven by corpus-based failure analysis. This is openly accessible via a community editable rule base.
The remainder of our paper is as follows. In Section 2., we provide descriptions of timexes according to modern standards and summarise previous approaches to timex normalisation in Section 3. We describe our system in Section 4. and present a comparative evaluation of state-of-the-art timex normalisation systems with a new gold-standard corpus in Section 5. before concluding.

2. Background

2.1. Temporal Expressions and Normalisation

Timex processing consists of recognising temporal expressions in text, as well as classifying and normalising them. The normalisation subtask consists of obtaining the absolute value of a timex regardless of the linguistic expression used. Example 1 shows the normalisation of two timexes.

(1) a. He was born in June $1983_{1983-06}$.
b. He was born in 06/$1983_{1983-06}$.

The timexes underlined in (1a) and (1b) normalise to the same value (*1983-06*). All semantically equivalent timexes encode to the same value.
Currently, the standard temporal annotation scheme is TimeML[2] (Pustejovsky et al., 2003a), which includes a specification of the TIMEX3 standard. According to this scheme, the normalised value of a timex is commonly expressed in one of the following notations: (i) a Gregorian calendar time or date formatted in the ISO 8601 standard as in (2a), or (ii) a period formatted as `P`, a `number` and an abbreviated `temporal unit` as in (2b).

(2) a. October 2012: `2012-10`
b. Two weeks: `P2W`

The complexity of the task comes from the variability of language for expressing time and also the fact that there are timexes whose accurate interpretation depends on other contextual and linguistic features – see Example (3).

(3) a. Monday
b. two days after

In (3a), we need the utterance or document creation time (DCT) and the tense to obtain the normalised value. In (3b), we need to refer to a previously-mentioned temporal expression to perform normalisation.

2.2. Timex Normalisation Taxonomy

Establishing a taxonomy of timexes is a useful first step in a "divide and conquer" approach to the normalisation problem. Most analyses agree that timexes can be:

- **Explicit, absolute, or self-contained**: These can be directly translated to a particular granularity date/time.
- **Implicit, relative, or context-dependent**: These need the document creation time (deictic) or a previously mentioned temporal reference/anchoring (anaphoric) to obtain a explicit date/time.
- **Durative**: Describing a bounded interval (or duration) that is not inherently anchored to a timeline.
- **Set or frequency**: Regularly recurring times, such as *"every Christmas"* or *"each Tuesday"*.
- **Vague**: generic mentions like *"recently"* or *"today"* in *"today's fashions"*; see TIDES standard Section 4.6 (Ferro et al., 2005).

The information and reasoning required to type and interpret temporal expressions is complex. One must rely on contextual clues, world knowledge and discourse anaphora in order to correctly perform timex normalisation.

3. Previous Work

There have been many previous approaches to the timex normalisation task. We first describe early systems which laid the foundation for timex normalisation, and then state-of-the-art systems focusing on those involved in the last international evaluation, TempEval-2 (Verhagen et al., 2010).

3.1. Pre-TIMEX3

One of the first relevant approaches to the normalisation task was TempEx (Mani and Wilson, 2000) later extended and released as GUTime (Verhagen et al., 2005). This was an early approach to the robust temporal processing of news. It defined an annotation scheme and a system to recognise and normalise timexes, excluding durations (e.g., *"two years"*), generics (e.g., *"today's youth"*) and fuzzy expressions (e.g., *"in a few hours"*). Its normalisation component distinguished explicit (self-contained or SC) and implicit (context-dependent or DP) timexes. Furthermore, within the implicit ones they differentiated those relative to the DCT and those relative to a previously mentioned timex (*temporal focus*). For normalising, the system used the following features relative to a timex: words, the reference time (DCT or focus) and governing tense.
In the TERN 2004 evaluation, the Chronos system (Negri and Marseglia, 2004) reached the highest performance. This system followed the TIMEX2[3] annotation guidelines. Similar to GUTime, the approach differentiated absolute and relative timexes. Again, these were divided into those relative to DCT and those relative to a previous temporal reference.
TERSEO (Saquete et al., 2006) is a later TIMEX2 system that improves upon Chronos. It accounts for period and fuzzy expressions. The system includes a FOL-based rule syntax and a method to automatically extend rules to other languages.
DANTE (Mazur and Dale, 2007) also uses rules and follows TIMEX2. The differentiation of underspecified timexes and the definition of their local semantics is the main contribution of this work.
TimexTag (Ahn et al., 2005) is another rule-based approach to normalisation which follows TIMEX2 specifications. This defines a taxonomy consisting of points, which

[2] See `http://timeml.org/`.

[3] See `http://fofoca.mitre.org/`.

includes explicit, deictic (relative to DCT) and anaphoric (relative to previous reference) timexes, durations, vague points (e.g., *"in the past"*) and recurrences (e.g., *"each Sunday"*). TEA (Han et al., 2006) uses rules but follows the recent TIMEX3 standard. Both TimexTag and TEA rely on hierarchical constraint satisfaction.

3.2. State of the art

Recently, systems have focused on the TIMEX3 standard, which was also used in TempEval-2. The best systems in the normalisation task were rule-based.

HeidelTime (Strötgen and Gertz, 2010) performed recognition and normalisation with rules, and catered for explicit, durative, implicit (relative to DCT) and relative timexes. Its regex-based rules use an internal symbol set encoding for temporal concepts such as months and calendar events. This system had the best normalisation performance over the TempEval-2 expressions that it recognised (85%).

TRIPS/TRIOS (UzZaman and Allen, 2010) had a data-driven recognition component, with rule-based timex typing and normalisation.

TIPSem (Llorens et al., 2010) implemented a hybrid strategy for normalisation. Firstly a learned classifier determines the normalisation type (explicit, relative to DCT, relative to previous reference, duration, set or vague). Then, handcrafted rules based on pattern matching are applied depending on normalisation type.

Finally, although not part of the TempEval-2 exercise, TERNIP (Northwood, 2010) is a modern re-implementation of GUTime, using a system-independent rule base and sophisticated syntax.

From the described systems, the following are publicly available and will work with TIMEX3-annotated TimeML: TERNIP[4] (based on GUTime), TIPSem[5] and HeidelTime[6]. Therefore, these are the systems to which we will compare TIMEN (see Section 5.).

4. TIMEN Overview

We introduce our approach in this section. The overall operation of TIMEN is as follows. Firstly, the timex phrase to be normalised is selected together with some contextual information. Next this is converted into a symbolic representation using a knowledge base (**KB**). Rules are then matched against the representation. Finally, a normalised output is produced in TIMEX3 format.

The distinguishing characteristics of TIMEN are independence from other timex processing tasks, an open philosophy and multilinguality. Its architecture clearly separates: (i) the algorithms (source code) which conduct the task and (ii) the knowledge and rules necessary for the process.

4.1. TIMEN Library

We developed TIMEN as a resource that outputs a timex's normalised value given a timex, a set of features (a DCT, a time_ref and a tense) and an input language. The architecture is shown in Figure 1. To obtain the normalised value TIMEN makes use of the knowledge database (KB) and the rule database (rulebase).

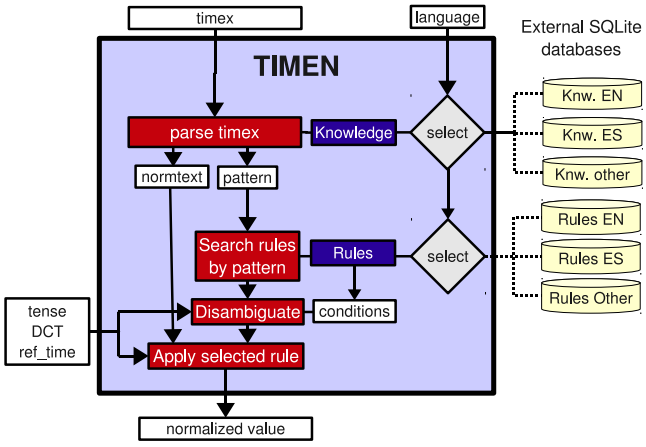

Figure 1: TIMEN Architecture

To better explain the processing flow of TIMEN, we describe the steps below using an example.

4.1.1. Input Data

Suppose that we run TIMEN with the following input data: timex is `October 25`, DCT is 2012-02-02, the tense is past and the language is English.

4.1.2. Symbolic Representation

Firstly, TIMEN parses the timex using the English KB (see Section 4.3.) to obtain a normalised text (normtext) and a pattern, which is the original text with certain phrases – such as weekday names – converted to language independent symbols.

The pattern is used to match rules in the rulebase, and the normtext is used to obtain the final normalised value from a simplified text (e.g., always in lower case, spelled numbers are translated to numbers, tokenisation where _ is the separator).

Example: For *October 25*, the pattern is *TMonth_Num*. This would be the same for any month followed by a number (e.g., Mar 1999). For *October 25*, the normtext is *october_25*.

4.1.3. Rule Matching

The next step is querying the rulebase with the pattern obtained. In cases where multiple rules match, TIMEN follows a disambiguation process. This consists of checking in order if the found rules have conditions and if so checking whether normtext matches them. The first rule found that matches the conditions (or does not have conditions) is applied.

Example:

In the current TIMEN rulebase, we find two rules for *TMonth_Num*. The first one refers to a month followed by a day and has the condition TOKEN(1)<32, which means that the word in position 1 (starting at 0) of normtext is a number and it is lower than 32. The second one refers to a month followed by a year and has no conditions. Since the rules are stored and retrieved in order of priorty, the first one will be always checked before the second.

[4] See `https://github.com/cnorthwood/ternip`.
[5] See `http://gplsi.dlsi.ua.es/demos/TIMEE`.
[6] See `http://dbs-projects.ifi.uni-heidelberg.de`.

For *october_25*, the condition of the first rule is matched. TIMEN applies the rule's expression – *DATE_MONTH_DAY(DCT,TOKEN(0),TOKEN(1))* – to obtain the TIMEX3 normalisation of a month followed by a day, by use of a built-in function (*DATE_MONTH_DAY*).
Below in the subsection dedicated to the knowledgebase and the rulebase (4.3.) we further explain the elements of the patterns and the syntax of the rules[7].

4.1.4. TIMEX3 Output

In order to apply the matched rule TIMEN will use the symbolic representation of the input text and the supplied input features.
Example: Information required at this point is tense, DCT and the values *october* and *25*. In this case, since the tense is past and the DCT is set to 2nd of February 2012, the normalised output will be *2011-10-25*.

4.1.5. Discourse Management

Since TIMEN is just a library we need an application which uses it and handles discourse-level information and document processing. In order to give to final users an example application, we developed TIMEN_CONSUMER.

4.2. Exmple Application Using TIMEN

TIMEN_CONSUMER is an example application developed to show how to integrate TIMEN in major projects. TIMEN_CONSUMER performs timex normalisation in two basic situations:

1. There is a timex without any context (e.g., *"October"*) and you want to know its normalised value.
2. There is a TimeML file where DCT and timexes are annotated and you want to add or update the normalisation values of the timexes.

As discussed above, a timex often cannot be normalised in isolation – contextual information such as temporal references and the tense of the verbs governing the timexes is often required. Therefore, to demonstrate the benefits of TIMEN over prior timex processing systems, we focus on the second situation.
Figure 2 shows the architecture of TIMEN_CONSUMER integrating the TIMEN library.
TIMEN_CONSUMER controls discourse-level information, saving previously normalised timexes in order to track reference time (Reichenbach, 1947) and also handling tenses which affect timexes. It takes a TimeML document containing delimited timexes as input and manages their presentation to the TIMEN library, which generates normalisations. Because TIMEN is available as a decoupled library, anyone may implement their own wrapper strategy for handling context; TIMEN_CONSUMER is provided as an example, to allow rapid standalone normalisation.

4.3. Knowledgebase and Rulebase

TIMEN relies on external knowledge, stored as symbolic or axiomatic representations. Here we describe the management of fixed language-specific knowledge and the rule format for temporal reasoning and normalisation.

[7] A complete technical reference to the knowledge base and rule syntax, as well as the syntax of rule conditions, is described at `http://timen.org/`.

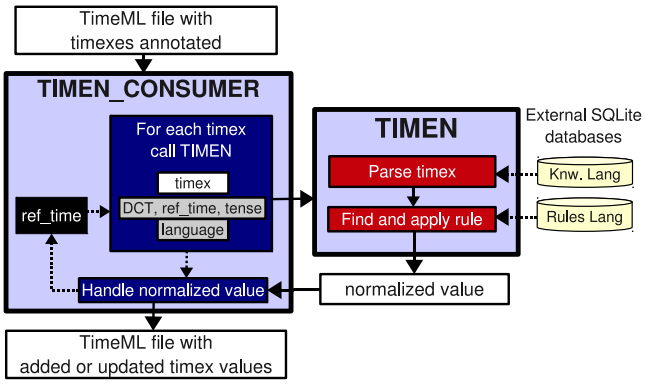

Figure 2: Architecture Overview

TIMEN includes an independent knowledge base and rulebase for each language stored in a user-modifiable format, outside of the processing logic. Based on the feature description of a timex, TIMEN normalises the timex using language knowledge (e.g. month name to month number mappings) and the rule databases.

4.3.1. Knowledgebase Construction

The **knowledgebases** are simple files which contain regular expressions for different time-related expressions. For example, the knowledgebase for English has an expression for months (*TMonth*) as shown in example (4).

(4) **TMonth** = (January|February|March|...|December) | (Jan|Feb|Mar|Apr|May|...|Sep|Sept|Oct|Nov|Dec)

TIMEN uses this knowledge to build language-independent representations from input timexes. For optimisation, the rulebases are maintained as static Java class files, automatically recompiled after modification.

4.3.2. Rule Storage

The **rulebases** are SQLite databases, which contain tables of rules. Rules operate on a priority and constraint-satisfaction basis. Rules chosen for normalisation are those that match the timex's pattern, in order of priority, highest first. In the case that a rule has conditions, it can only be applied if the timex satisfies them.

4.3.3. Rule Syntax

Some of the basic constants and functions which can be used in the rules are the following.

Constants: This are elements that are replaced by the corresponding value set at TIMEN initialisation time.

- **DCT**: The value of the document creation time.
- **REFTIME**: The value of the current reference time.
- **DCTYEAR**: The four-digit year of the DCT.
- **DCTMONTH**: The two-digit month of the DCT.
- **DCTDAY**: The two-digit day of the DCT.

Functions: These are elements used to calculate and print different normalisation values. Examples follow.

- **"STRING"**: Prints any quoted text string.
- **TO_YEAR(number)**: If a number does not have four digits the missing digits are guessed taking into account DCT and tense. For example, if the tense is past and the DCT is in 2012, TO_YEAR(99), this function will return `1999`. If the tense is future it will return `2099` instead.
- **DATE_MONTH(date, month)**: Returns a date using the tense given a reference date and a month. For example, if the tense is future, DATE_MONTH(2012-02-02,october) returns `2012-10`.
- **ADD(date, granularity, number)**: Outputs the date resulting of adding the number in the corresponding granularity. For example, ADD(2012-03-01,day,-1) equals `2012-02-29`.

The complete rule syntax as well as the condition syntax is included in TIMEN documentation[8].

id	pattern	rule_type	rule	rule_cond
1	TWeekday	implicit_d	DATE_WEEKDAY(DCT,PAT(0))	
2	TMonth_Num	implicit_d	DATE_MONTH_DAY(DCT,PAT(0),PAT(1))	PAT(1)<32
3	TMonth_Num	explicit	TO_YEAR(PAT(1));"-";TO_MONTH(PAT(0))	
4	TMonth	implicit_d	DATE_MONTH(DCT,PAT(0))	
5	Num	explicit	TO_YEAR(PAT(0))	
6	today	implicit_d	DCTYEAR;"-";DCTMONTH;"-";DCTDAY	
7	yesterday	implicit_d	ADD(DCT,day,-1)	
8	tomorrow	implicit_d	ADD(DCT,day,1)	
10	Num_TUnit_a	implicit_d	ADD(DCT,PAT(1),NEGATIVEINT(PAT(0)))	
12	now	vague	"PRESENT_REF"	
20	Num_TUnit	durative	"P";TO_PERIOD(PAT(0),PAT(1))	
21	TUnit	durative	"P";TO_PERIOD(1,PAT(0))	
50	this_TUnit	implicit_d	ADD(DCT,PAT(1),0)	

Figure 3: Snapshot of rules (here *PAT()* is *TOKEN()*).

Note that a rule might consist of one or more constants or functions separated by the semi-colon character. Furthermore these can be combined, e.g., ADD(DCT,TOKEN(1),TOKEN(0)). To illustrate some real rules, a screen-shot of the rulebase containing some actual rule entries is shown in Figure 3. The example rule used in Section 4.1. can be seen here.
TIMEN and TIMEN_CONSUMER have been made available on-line, at the TIMEN website. This website also serves as an interface for the community integration of the resource explained in the next section.

4.4. Community Integration

In this section, we describe how we manage the collaborative community-led management of TIMEN's normalisation rulebase. Our approach to timex interpretation relies on the premise that the normalisation problem can only be completely solved with some application of hand-crafted rules. A universal tool needs a hugely comprehensive rulebase; one that cannot be constructed in a short amount of time or by a small group of people.
To overcome this, TIMEN's collaborative rule repository can be edited by any interested party. Modification and addition of rules must be enabled in a way that ensures quality. Each rule includes a full TIMEX3 annotation that shows a

[8] See `http://timen.org`.

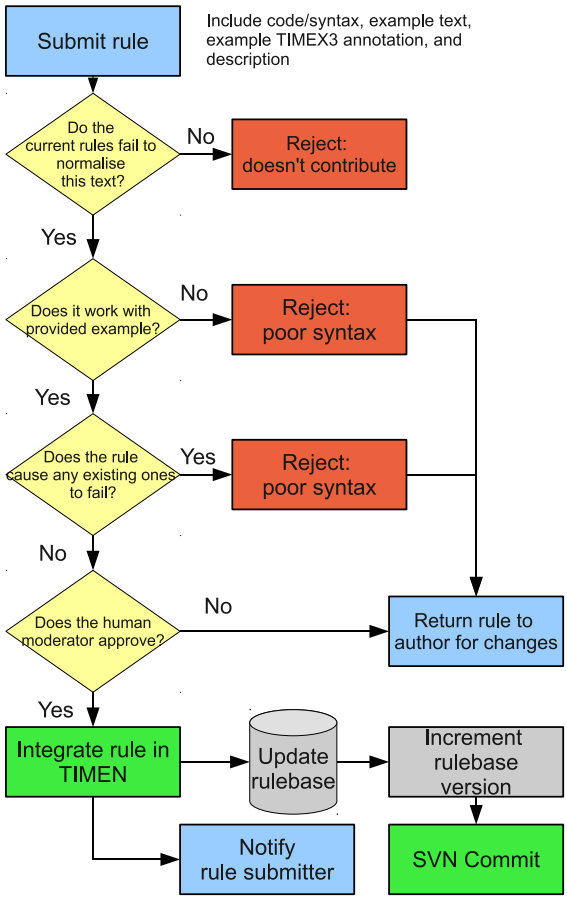

Figure 4: Moderation of new entries for TIMEN

timex in context and also its type and value. Using this, each rule can be verified. When a rule is to be modified or added, a testing component, which is part of TIMEN, verifies the rule set and highlights failing or broken rules.
We enable the community aspect of TIMEN with a website that lets users submit/modify rules, using:

- **Rule ID:** a numeric id for the rule.
- **Rule pattern:** the pattern to match.
- **Rule type:** the type in the taxonomy.
- **Rule code:** the encoded rule.
- **Rule conditions:** conditions if any.
- **Rule priority:** the importance of the rule (optional).
- **Language:** a two-letter language code representing which language rulebase this entry is destined for.
- **Example timex:** a sentence containing a TIMEX3 tag, that is normalisable using the rule. The normalised text and the annotated TIMEX3 value are compared to validate the rule's operation.

We manually moderate new rules, but they will be automatically rejected if the rule reduces performance on a predefined test set, the rule does not work for the example timex or the example can already by normalised by the existing ruleset. The process for moderating and accepting new rules is detailed in Figure 4. Rule priority is dealt with in more detail in the technical documentation on the TIMEN website.
TIMEN also includes an update mechanism which can au-

tomatically download new rulebase and knowledgebase entries into an existing installation. The website used for this process is based on Wordpress, and new rules use the "posts" mechanism, allowing us to re-use the moderation and comments facilities without building a new management system from scratch.

5. Evaluation

In this section, we evaluate TIMEN using an existing dataset that is commonly used for normalisation evaluation, and using a newly created corpus which we also introduce. The evaluation of TIMEN has two primary goals: (i) measuring the performance of TIMEN itself over gold TimeML annotated data, and (ii) measuring whether or not the application of TIMEN over currently available timex processing systems leads to an improvement of their performance in normalisation.

5.1. Development Data

The evaluation in this section is intended to reflect results based on resources that other state-of-the-art normalisation systems have. Therefore, the development dataset consists of the TimeBank (Pustejovsky et al., 2003b) and AQUAINT TimeML corpora plus the TempEval-2 English training TIMEX3 annotations.

5.2. TIMEX3 Evaluation Datasets

There are two evaluations dataset: the TempEval-2 test dataset and a newly created dataset (TimenEval). Both TIMEN and, possibly, other existing systems, have developed their normalisation rule sets using the TempEval-2 training dataset, a part of TimeBank 1.2 corpus[9].

5.3. New Dataset: TimenEval

For TIMEN evaluation, not only do we evaluate using a well-known prior dataset, we also create and include a new resource for timex normalisation evaluation. Most existing systems have been built using the same datasets for development – TimeBank, AQUAINT and TempEval-2. We therefore generated new data to see how the systems (including TIMEN) perform on unseen timexes. The dataset is intended to focus on diversity of both expression of time (e.g. input text) and value (normalised value). To this end, it contains a significant amount of non-newswire material. Vital statistics of the datasets are summarised in Table 1.

Test set	Docs	Words	Timexes	IAA
TempEval-2	9	5.5k	81	0.89
TimenEval	9	7.9k	214	0.91

Table 1: Evaluation Corpora Statistics. IAA here is strict extent + timex value; extended IAA figures are included with the TimenEval dataset, available via the TIMEN site.

For the new dataset (TimenEval), we took care to achieve a good distribution of dates, times, durations and sets. Two annotators manually selected English test documents from the TAC KBP Source Collection[10]. Annotations were validated both with the CAVaT TimeML checking tool (Derczynski and Gaizauskas, 2010) and also via XML Schema. TempEval-2's test data set had only 6 `TIME`s and no `SET`-type timexes; TimenEval improves notably on both these counts (43 *TIME*s and 16 *SET*s). TimenEval is available at `http://www.timen.org/`.

5.4. TIMEN Performance

For the first goal, we run TIMEN over the gold annotations of both the test set in the TempEval-2 evaluation and our newly created dataset. Our initial evaluation measures intrinsic normalisation accuracy using gold-standard timex annotations. Table 2 shows the obtained results.

Test set	Recall	Normalisation accuracy
TempEval-2	(1.0)	0.90
TimenEval	(1.0)	0.68

Table 2: TIMEN performance using gold-standard timex extents

Recall 1.0 reflects that these results apply to all timexes in the datasets, since TIMEN does not do any timex recognition.

So far, the current TIMEN version has only 76 rules covering only the most common timexes. Furthermore, TIMEN_CONSUMER does not use sophisticated NLP such as syntactic or semantic parsing but just tense recognition and timex tracking. With this young and incomplete rule set we obtained high results – 90% in TempEval-2 and 68% in TimenEval.

A priori, these results seem comparable to those obtained by the best normalisation system in TempEval-2 (e.g., HeidelTime: 0.85). However, because HeidelTime only recognised 86% of all timexes in the TempEval-2 test set, that 0.85 means that HeidelTime normalised correctly only 85% of the 86% of timexes it was exposed to. Results are not thus comparable and we can therefore only assert with full certainty that HeidelTime normalised correctly at least 0.73 ($0.85 * 0.86$) out of the total timexes.

To conduct a fair, comparable and rigorous evaluation, we carried out the second experiment detailed below, in which each system is evaluated and combined with TIMEN to detect if TIMEN gives an improvement in the normalisation performance – which is its goal.

5.5. Using TIMEN with other systems

For the second goal, measuring the performance change TIMEN offers over other systems' normalisation components, we compare the performance of three other publicly-available state-of-the-art timex processing systems (i.e., TIPSem, TERNIP, HeidelTime); firstly with their own normalisation code's TIMEX3 values and secondly with those supplied by TIMEN. Specifically, we run these systems over a dataset and measure their own normalisation performance. Then, we substitute their normalisation component with TIMEN and re-evaluate, using their recognised timex extents as input. This is made possible by

[9] See `http://timeml.org/site/timebank/` for TempEval dataset downloads.

[10] LDC ref. LDC2010E12, *TAC 2010 KBP Source Data*.

TIMEN_CONSUMER's support for operation as a drop-in component that can take existing TIMEX3 annotations and overwrite only the attributes relevant to normalisation.

We carried out the described evaluation over both the TempEval-2 test data and the TimenEval dataset, which, unlike the TempEval-2 dataset, can be guaranteed not to have been used to develop the evaluated systems or TIMEN. Tables 3 and 4 show the results obtained for each dataset. Performance is scored as in TempEval-2; that is, "Extents" is the F1 measure of strict extent detection, and the normalisation scores are the percentage of correct TIMEX3 values determined for the subset of detected timexes.

System	Extents	Internal norm.	TIMEN norm.	ER
TIPSemB	0.94	0.83	0.89	+35%
HeidelTime	0.86	0.94	0.94	+0%
TERNIP	0.85	0.76	0.92	+66%

Table 3: Systems evaluation over TempEval-2 test dataset. Abbreviations: ER (Error reduction)

Results for TempEval-2 data are shown in Table 3. Given timexes with extents determined by each system, TIMEN performs better than built-in normalisation components in all cases except HeidelTime's, where it matches performance. Bear in mind that HeidelTime has been actively developed long after TempEval-2 with access to this dataset, just as TIMEN has.

System	Extents	Internal norm.	TIMEN norm.	ER
TIPSemB	0.51	0.57	0.67	+23%
HeidelTime	0.70	0.72	0.74	+7.1%
TERNIP	0.73	0.70	0.72	+6.6%

Table 4: Systems evaluation over TimenEval. Abbreviations: ER (Error reduction)

Results on the TimenEval dataset are given in Table 4. This evaluation treats normalisation equally, as no system has seen the dataset before. Again, TIMEN provides a visible performance boost in all cases, even over HeidelTime's normalisation. It is interesting to see an increase here, as HeidelTime uses an integrated recognition and normalisation ruleset, and might not be expected to recognise timexes that it could not normalise. TIPSemB's low results are due to the variable quality of the text in the dataset. Timex annotations are not as structured as those in TempEval-2, but instead include some noise (e.g. newline breaks in timexes).

The lack of variety in genre in existing TimeML corpora, which are all newswire, also makes it harder for existing approaches based on machine learning to recognise timexes in TimenEval. To improve recognition, re-training on informal/unformatted text is required.

5.6. Error analysis and discussion

Development of the gold standard dataset was hard, particularly because of variation in available standards. ISO-8601, TIDES TIMEX2 and TimeML TIMEX3 all contribute to the normalisation value format. Example cases include the treatment of negative dates, where (for historic reasons) TIDES and ISO-8601 diverge; for date delimitation, where ISO-8601 permits a variety of schemes, TIMEX2 and TIMEX3 are non-specific, but all existing tools and resources use hyphenated dates; and treatment of generic temporal pronouns, such as in *"this time"*. Our experience suggests that future clarification and improvement of the TimeML TIMEX3 guidelines is warranted (especially in terms of adding examples and of dealing with SETs), perhaps as a contribution to ISO-TimeML.

The normalization granularity (level of detail) of some timexes is often complex. For example, when discussing the third quarter of 1989 with a DCT of 1989, *"a year ago"* could be interpreted as 1988-Q3 instead of 1988. In the evaluation datasets, there are ambiguities in this kind of expression.

Words that describe times briefly using contextual clues were also hard to normalise, as in *"Why 10am? Why not twelve, or two, or four?"*. Complex cases were also difficult, such as with *"Every 2 weeks at 16:00 on Saturday"*.

Finally, unusual phrases were difficult to interpret. These should be the easiest category of all for which to add rules to a communal resource. They include items such as *"Purim"*, *"the intercalary day"* and *"Mid-Autumn Festival"*. Gazetteers and other lists of holiday events contain explicit rules for resolving a holiday name to a date, given a year. Combing through these resources and adding them to TIMEN will improve our normalisation coverage.

6. Conclusion

We have presented an open and independent state-of-the-art tool for timex normalisation: TIMEN.

Its performance results show improvement in normalisation over recent approaches, demonstrating that TIMEN is a new effective resource for normalisation regardless of timex recognition approach. Furthermore, it removes the cost of re-developing a timex normalisation system. This saves time and favors the improvement of the resource via community contribution and evaluation. This makes TIMEN an extensible and reusable tool, finally overcoming the boundary that all previous timex normalisation systems have suffered from at the end of their development.

Regarding future work, the central effort is the ongoing extension and refinement of TIMEN's rule base. We encourage community participation in this. Improving performance through building up the rule base, will make TIMEN a viable long-term choice for timex normalisation.

We plan to extensively evaluate TIMEN over more data and publish a high-coverage normalisation resource not limited to English but including other languages such as Spanish, Chinese, Italian and Danish. As the TempEval-2 dataset includes many languages, initial construction of these rule bases can be data-driven. The nature of our framework reduces the barrier to adding new languages and to contribute to the normalisation of those languages, as it uses only a simple purpose-built rule syntax.

Acknowledgments

Leon Derczynski would like to acknowledge the UK Engineering and Physical Science Research Council's support in the form of a doctoral studentship. This paper has been also supported by the Spanish Government, in projects TIN-2009-13391-C04-01, MESOLAP TIN2010-14860, PROMETEO/2009/119 and ACOMP/2011/001.

7. References

David Ahn, Sisay F. Adafre, and Maarten de Rijke. 2005. Towards Task-Based Temporal Extraction and Recognition. In *Annotating, Extracting and Reasoning about Time and Events*.

Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. 2007. On the Value of Temporal Information in Information Retrieval. *SIGIR Forum*, 41(2):35–41.

Naomi Daniel, Dragomir Radev, and Timothy Allison. 2003. Sub-event based multi-document summarization. In *HLT-NAACL Text summarization workshop*, pages 9–16. ACL.

Leon Derczynski and Robert Gaizauskas. 2010. Analysing Temporally Annotated Corpora with CAVaT. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 398–404, Valletta, Malta.

Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson. 2005. TIDES 2005 Standard for the Annotation of Temporal Expressions. Technical report, MITRE.

Benjamin Han, Donna Gates, and Lori Levin. 2006. From language to time: A temporal expression anchorer. In *Proceedings of the 13th International Symposium on Temporal Representation and Reasoning (TIME)*, pages 196–203, Washington, DC, USA. IEEE.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Xuansong Li, Kira Griffit, and Joe Ellis. 2011. Overview of the TAC2011 Knowledge Base Population Track. In *Proceedings of the Text Analytics Conference*, Gaithersburg, MD, USA.

Hector Llorens, Estela Saquete, and Borja Navarro-Colorado. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291. ACL.

Hector Llorens, Estela Saquete, and Borja Navarro. 2011. Syntax-Motivated Context Windows of Morpho-Lexical Features for Recognizing Time and Event Expressions in Natural Language. In *Natural Language Processing and Information Systems*, pages 295–299. Springer.

Inderjeet Mani and George Wilson. 2000. Robust temporal processing of news. In *ACL Annual Meeting*, pages 69–76, NJ, USA. ACL.

Pawel Mazur and Robert Dale. 2007. The DANTE temporal expression tagger. In *Proceedings of the 3rd Language and Technology Conference (LTC)*.

Matteo Negri and Luca Marseglia. 2004. Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004. Technical report, Information Society Technologies.

Chris Northwood. 2010. TERNIP: Temporal Expression Recognition and Normalisation in Python. Master's thesis, University of Sheffield.

James Pustejovsky, José M. Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *IWCS-5*.

James Pustejovsky, Patrik Hanks, Roser Saurí, A. See, Robert Gaizauskas, Andrea Setzer, Dragomir R. Radev, Beth Sundheim, David Day, Lisa Ferro, and M. Lazo. 2003b. The TimeBank Corpus. In *Corpus Linguistics*, pages 647–656, Lancaster, UK.

Hans Reichenbach. 1947. The tenses of verbs. In *Elements of Symbolic Logic*, pages 287–298. The Macmillan Company, New York.

Estela Saquete, Rafael Muñoz, and Patricio Martínez-Barco. 2006. Event ordering using TERSEO system. *Data Knowledge Engineering*, 58(1):70–89.

Estela Saquete, José Luis Vicedo González, Patricio Martínez-Barco, Rafael Muñoz, and Hector Llorens. 2009. Enhancing QA Systems with Complex Temporal Question Processing Capabilities. *Journal of Artificial Intelligence Reresarch (JAIR)*, 35:775–811.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

Naushad UzZaman and James F. Allen. 2010. TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 276–283, Uppsala, Sweden. ACL.

Marc Verhagen, Inderjeet Mani, Roser Saurí, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, and James Pustejovsky. 2005. Automating temporal annotation with TARSQI. In *ACL*, pages 81–84, NJ, USA. ACL.

Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden. ACL.