

# Integer-Programming Ensemble of Temporal-Relations Classifiers

Catherine Kerr · Terri Hoare  
Paula Carroll · Jakub Mareček

Received: date / Accepted: date

**Abstract** The extraction of temporal events from text and the classification of temporal relations among both temporal events and time expressions are major challenges for the interface of data mining and natural language processing. We present an ensemble method, which reconciles the outputs of multiple heterogenous classifiers of temporal expressions. We use integer programming (IP), a constrained optimisation technique, to improve on the best result of any individual classifier by choosing consistent temporal relations from among those recommended by multiple classifiers. Our ensemble method is conceptually simple and empirically powerful. It allows us to encode knowledge about the structure of valid temporal expressions as a set of constraints. It obtains new state-of-the-art results on two recent natural language processing challenges, SemEval-2013 TempEval-3 (Temporal Annotation) and SemEval-2016 Task 12 (Clinical TempEval), with F1 scores of 0.3915 and 0.595 respectively.

## 1 Introduction

The extraction of expressions related to time (temporal expressions), expressions related to events (temporal events), and understanding relations among them are major challenges in data mining, information retrieval, and natural language processing. Applications range from compiling situational-awareness reports in disaster management (Ashish et al., 2008), monitoring the spread

---

C. Kerr, T. Hoare, and P. Carroll  
University College Dublin  
Belfield, Dublin 4, Ireland  
paula.carroll@ucd.ie

J. Marecek  
IBM Research – Ireland  
B3 F14 IBM Campus Damastown, Dublin 15, Ireland  
jakub.marecek@ie.ibm.com

of disease (Elkin et al., 2012), distinguishing causes and effects in clinical applications (Huang and Lu, 2016; Styler IV et al., 2014), automating trading strategy responses to news reports in computational finance (Nuij et al., 2013; Benetka et al., 2017), machine reading in military intelligence (Bier et al., 2008), to question-answering in virtual assistants and other web-scale tasks (Madhavan et al., 2007; Movshovitz-Attias et al., 2010).

Often, much is at stake. Consider the task of understanding and ordering events in disaster management. Traditionally, staff of emergency response agencies monitor print and broadcast media, the web, and internal information systems to compile situational awareness reports and coordinate responses to emergency situations. Much of this data is in unstructured natural-language, and often multiple media refer to the same event, but possibly in different ways. Further complications arise due to data protection rules, which bar the naming of individuals, and the inherent ambiguity in natural languages. Consider, for example, the reports “three people died in the past hour near X” and “five people have died over the past two days near X”. We might hope that at most five people died altogether at that location, if the hour is a subinterval of the two days, but this deduction requires a temporal ordering of the two events. Monitoring activities, which require such deduction, are very costly, not only in terms of direct costs, but also in terms of opportunity costs, e.g., putting more “boots on the ground”. Hence, the automation of the compilation of situational-awareness reports was suggested as one of the first applications for temporal-relations processing by Ashish et al. (2008).

In medicine, there are also challenges in understanding the sequencing of events in clinical narratives (Elkin et al., 2012; Huang and Lu, 2016; Styler IV et al., 2014). Clinical narratives are also used to monitor the spread of disease (Elkin et al., 2012). Consider, for example a statement: “before he took A, everything had seemed normal. Then, suddenly, he experienced B, and took C”. We might suspect that condition B occurred in response to taking A, but this inference requires a temporal ordering of the three events, which is not the same as the order of appearance of the events in the text. Automating the understanding of patients’ medical history records could improve diagnostics and survival rates for individual patients, as a sequence of symptoms captured by a clinical narrative may help to understand a disease. Once the relative ordering of events is recognised, it may also allow for secondary research across patient records. Databases of narratives such as the Adverse Event Reporting System of the U.S. Food and Drug Administration (Tatonetti et al., 2012), and social media (Comfort et al., 2018) provide textual corpora which are at the core of considerable business activity, including IBM Watson Health.

In computational finance, rapid response to reports of financial results, mergers and acquisitions, or pronouncements of analysts can lead to substantial improvements in automated trading strategies Nuij et al. (2013). In such applications, it is important not to overreact and respond to one event multiple times, and to recognise that the same event may have been reported differently in multiple sources. While much of the work in finance remains unpublished, we refer to Benetka et al. (2017) for a recent thoughtful study.

## 1.1 The Temporal Reasoning Challenge

Throughout many such applications, we need to consider complete passages of text, because individual sentences may be ambiguous. The extraction of time expressions and expressions related to events is often the first step. For example, Sauri et al. (2005) focus on the identification of temporal events in texts.

The second step is temporal-relation classification, i.e., the association of events with temporal expressions and the identification of relations among temporal expressions such as deciding which event came first. A number of approaches have been presented by, e.g., Chambers (2013); Bethard (2013); Laokulrat et al. (2013); Leeuwenberg and Moens (2016); Caselli and Morante (2016); Chikka (2016); Grouin and Moriceau (2016); Khalifa et al. (2016); Cohan et al. (2016), and Lee et al. (2016).

## 1.2 Our Focus

The estimates of temporal relations from different classifiers, are generally not consistent (Bhattacharya and Getoor, 2007). Deriving a single *consistent* estimate of the temporal-relation classification is a computationally hard problem in general.

Our main contribution is an integer-programming (IP) ensemble framework which enforces temporal consistency rules for the temporal-relations classification problem. While other ensemble methods may use performance metrics as voting weights, they do not have a mechanism to enforce consistency rules or to exploit the structure of temporal relationships, but rather learn the weights by manipulating the data to generate and then ensemble multiple classifiers (Dietterich, 2000). IP is a constrained optimisation technique that decides solution values from among a set of alternatives by maximising an objective function subject to a set of constraints. In this paper the IP ensemble decides the best consistent temporal labellings while allowing the diversity of labellings to be exploited. It provides a single interpretation of the temporal relations that reconciles conflicts between the recommendations of the individual heterogeneous classifiers. The solution of the IP ensemble therefore provides a more consistent interpretation of the temporal information than weighted averages of the individual classifiers.

We demonstrate through empirical experiments that our IP ensemble approach is computationally efficient in practice. We test our IP formulation on a series of natural language processing (NLP) competitions, known as the Temporal information extraction challenge (TempEval) (Verhagen et al., 2009), and Semantic Evaluation (SemEval). We generate multiple estimates of the temporal event labellings by using several different classifiers provided to us by participants in the challenges. The individual classifiers employ a variety of machine-learning techniques including support vector machines and methods inspired by maximum-entropy, which yield very different results. We use the

IP ensemble to reconcile the outputs of the classifiers and decide a final consistent labelling. The IP ensemble maximises a performance metric subject to a full set of consistency constraints. We evaluate several metrics such as  $F_1$  scores.

Our IP framework synthesises concepts from a number of areas, which are introduced in Section 2. We describe our IP ensemble framework in Section 3, we present computational results and analysis in Section 4 which show that the IP ensemble yields an improvement over any of the individual classifiers. Lastly, we present our conclusions in Section 5.

## 2 Temporal Reasoning in Natural Language Processing

In this section we provide a brief introduction to a number of topics including Allens interval algebra for temporal reasoning, the temporal event markup language TimeML, the use of temporal event graphs to model TimeML annotated text, and decision making by integer programming. Even readers familiar with these topics may find the content serves as a useful refresher.

In order to automate temporal event analysis, we need to extract and analyse the temporal expressions and descriptions of events in textual reports and link the events appropriately. Consider the challenge to identify and order the temporal events in the following sample sentence from the TempEval-3 test data: “President Barack Obama arrived in refugee-flooded Jordan on Friday after scoring a diplomatic coup just before leaving Israel when Prime Minister Benjamin Netanyahu apologized to Turkey for a 2010 commando raid that killed nine activists on a Turkish vessel on a Gaza bound flotilla.”

We can identify several events: Israel *raided* a Turkish vessel, *killed* nine activists, Benjamin Netanyahu *apologized*, Barack Obama *scored* a coup, *left* Israel for *Jordan*. Further, we can identify that the events happened in this order, possibly with leaving Israel happening at the same time as entering Jordan, and also possibly with the apology and the scoring of the diplomatic coup coinciding. We can also identify two temporal expressions (2010, Friday) and associate them with the respective events. While this may be trivial for humans, it is not trivial for a computer. Indeed, it may not even be clear how to formalise the problem.

Over the past half century, several frameworks have been designed to describe both qualitative and quantitative relations between time points and time intervals, as well as combinations of these (Daykin et al., 2016). A calculus, or relational algebra, for ordering the relationship of temporal events gives us a way to formalise a set of (temporal) rules. The application of the global rules of such a calculus on the full set of temporal relations provides a necessary and sufficient condition for consistency across the relations in a news-feed document (Ladkin, 1990), and allows us to produce meaningful results for the temporal reasoning problem.

In the following sections, we describe one such algebra (Section 2.1) and show how it is used in temporal reasoning with the support of a temporal

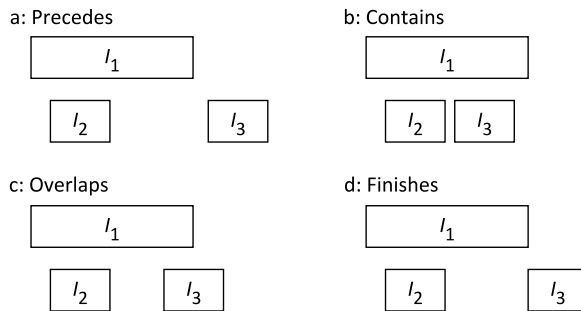
markup language, TimeML (Section 2.2) and graphical models (Section 2.3) in TempEval competitions (Section 3.2). We also describe integer programming, the decision making tool used in our ensemble (Section 2.4).

## 2.1 Allen's Interval Algebra

Allen's interval algebra is a popular formalism for reasoning about the relationships between time intervals Allen (1983). Allen (1983) provides an algebra of binary relations on intervals where time is represented by intervals and there are thirteen mutually exclusive dyadic relations between pairs of intervals. These thirteen relation types (*reltypes*) are Precedes, Meets, Overlaps, Starts, Finishes, and During, their converses (Preceded by, Met by, Overlapped by, Started by, Finished by and Contains), and the equals relation. The thirteen basic reltypes precisely characterise the relative start and end points of two temporal intervals.

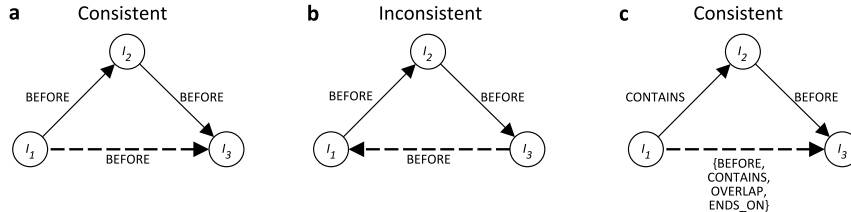
Allen also describes the concepts of temporal path consistency and transitive closure of the network of relations between intervals, concepts which were extended by Ladkin (1990). Between any two temporal intervals exactly one of the relations holds. Transitivity properties are used to determine which relations may hold between pairs of intervals based on known relations between other intervals. The problem of deciding which reltype holds can be represented as a directed graph of the interval algebra. Paths are consistent if there are no impossible relations among all triples of nodes in the graph.

Consider a simple example, where  $I_1$  contains  $I_2$ , and  $I_2$  precedes  $I_3$ . What can be said about the relationship type (reltype) between  $I_1$  and  $I_3$ ? Figure 1 illustrates the four options for completing the transitive closure of the interval network within Allen's Interval Algebra, with time on the horizontal axis in each subfigure.



**Fig. 1** An example of possible classifications of reltypes between  $I_1$  and  $I_3$ , considering  $I_1$  contains  $I_2$ , and  $I_2$  precedes  $I_3$ .

In Figure 2, we illustrate the inference. Solid arcs show the known reltypes, while dashed arcs represent the reltype we are attempting to infer. In Subfigure 2.a, inferring the reltype *Precedes* is consistent, while in Subfigure 2.b, it is not. Subfigure 2.c shows that choosing one of the listed reltypes is consistent, but we must decide which is the most appropriate labelling of the dashed arc.



**Fig. 2** Examples of reasoning with Allen’s Interval Algebra.

Allen (1983) describes a polynomial time constraint propagation algorithm to compute an approximation of the strongest implied relation for each pair of intervals. He demonstrates that the algorithm is correct and never infers an inconsistent path between a pair of intervals. However, Allen also shows that the algorithm is incomplete and does not make all the inferences, i.e., complete the transitive closure. Nebel and Bürckert (1995) discuss the computational challenge and tractability of algorithms for the transitive closure of interval algebra networks. They note that the problem is NP-Complete in the full algebra, but propose polynomial time algorithms for subsets of the algebra.

Allen encoded transitivity information in a composition table, so that reasoning in the interval calculus starts with a look-up in the table. The TempEval competitions use different subsets of Allen’s interval algebra depending on the task. Tables 1 and 2 show the relations composition tables for the two competitions, in the TimeML format. The left-most columns show the reltypes alongside a short code, e.g., ‘p’ for Before (which is the TimeML encoding of *Precedes* in Allen’s Interval Algebra). Rows represent  $I_1 \circ I_2$  and columns represent  $I_2 \circ I_3$ , where  $\circ$  represents any of the reltypes. The cell where the row and column intersect lists possible reltypes of  $I_1 \circ I_3$ . For example, given that interval  $I_1$  is before  $I_2$  and that  $I_2$  is before  $I_3$ , we can infer from Table 1 that  $I_1$  is before  $I_3$  as in Subfigure 2.a. This also tells us that  $I_3$  before  $I_1$  is inconsistent with the given information, as we saw in Subfigure 2.b.

Subsets of the reltypes can be used depending on the specific NLP task. The reltypes used in the SemEval-2016 clinical text challenge are shown in Table 2. Taking as an example  $I_1 \circ I_2$  is *c* (Contains) and  $I_2 \circ I_3$  is *p* (Before), as we saw in Subfigure 2.c, then  $I_1 \circ I_2 \circ I_3 = I_1 \circ I_3 \in \{p, c, o, bi\}$ , i.e., the composite relation is one of Before (*p*), Contains (*c*), Overlap (*o*), or Ends-on (*bi*, which is the TimeML attribute of Allen’s *Finishes*).

**Table 1** Transitivity relations in the news-feed challenge (SemEval-2013 TempEval-3 Task C)

	p	pi	o	oi	I	Ii	m	mi	s	si	f	fi
BEFORE (p)	p	.	p	p, d, o, m, s	p	p	p	p, d, o, m, s	p	p	p, d, o, m, s	p
AFTER (pi)	.	pi	pi, d, oi, mi, f	pi	pi	pi	pi, d, oi, mi, f	pi	pi, d, oi, mi, f	pi	pi	pi
INCLUDES (o)	p	pi, di, oi, mi, si	p, o, m	.	o	o	p	di, oi, si	o	di, o, fi	d, o, s	p, o, m
IS_INCLUDED (oi)	p, di, o, m, fi	pi	.	pi, oi, mi	oi	oi	di, o, fi	pi	di, o, fi	pi, oi, mi	oi	di, oi, si
IDENTITY (I)	p	pi	o	oi	I	I	m	mi	s	si	f	fi
SIMULTAN- EOUS (Ii)	p	pi	o	oi	Ii	Ii	m	mi	s	si	f	fi
IBEFORE (m)	p	pi, di, oi, mi, si	p	d, o, s	m	m	p	I, Ii, f, fi	m	m	d, o, s	p
IAFTER (mi)	p, di, o, m, fi	pi	d, oi, f	pi	mi	mi	I, Ii, s, si	pi	d, oi, f	pi	mi	mi
BEGINS (s)	p	pi	p, o, m	d, oi, f	s	s	p	mi	s	I, Ii, s, si	.	p, o, m
BEGUN_BY (si)	p, di, o, m, fi	pi	di, o, fi	oi	si	si	di, o, fi	mi	I, Ii, s, si	si	oi	di
ENDS (f)	p	pi	d, o, s	pi, oi, mi	d	f	m	pi	.	pi, oi, mi	d	I, Ii, f, fi
ENDED_BY (fi)	p	pi, di, oi, mi, si	o	di, oi, si	fi	fi	m	di, oi, si	o	di	I, Ii, f, fi	fi

**Table 2** Transitivity relations in the clinical-notes challenge (TempEval 2016 Task 12.)

Relation	p	pi	c	ci	o	b	bi
BEFORE (p)	p	.	p	p, ci, o, bi	p, ci, o, bi	p, ci, o, bi	p
AFTER (pi)	.	pi	pi	pi, ci, o, b	pi, ci, o, b	pi, ci, o, b	pi
CONTAINS (c)	p, c, o, bi	pi, c, o, b	c	c, ci, o	c, o	c, o	c, o
CONTAINS_INV (ci)	p	pi	.	ci	p, pi, o, b, bi	pi	p
OVERLAP (o)	p, c, o, bi	pi, c, o, b	c, o	ci, o	ci, o, b, bi	pi, c, o	p, c, o
BEGINS_ON (b)	p, c, o, bi	pi	pi	ci, o	pi, ci, o	pi	c, o
ENDS_ON (bi)	p	pi, c, o, b	p	ci, o	p, ci, o	c, o	p

## 2.2 TimeML Markup Language

Temporal relations and events in text can be tagged using a markup language such as TimeML. TimeML is a temporal information standard markup language (Pustejovsky et al., 2005; Saurí et al., 2009). TimeML uses four primary tag types:

- *EVENT* for temporal events;
- *TIMEX3* for temporal expressions;
- *SIGNAL* for temporal signals;
- *TLINK* to represent relationships between events and times or between events.

TimeML encodes the Allen Intervals and includes a fourteenth relation to include the converse of equals. Each event has a unique identifier (EVENT-ID) and is associated (by a TLINK) with some time expression (TIMEX).

Styler IV et al. (2014) propose an extension to capture the requirements for the annotation of temporal information within clinical narratives.

In the example of the previous subsection, the event of “President Obama’s arrival” is associated with the time expression of “Friday” by means of a TLINK. When we use  $E^i$  ( $T_i$ ) to denote the  $i^{th}$  event (time expression, respectively), the full TempEval-3 annotation of the sentence can be:

President Barack Obama *arrived*<sup>E1</sup> in refugee-flooded Jordan on *Friday*<sup>T1</sup> after *scoring*<sup>E2</sup> a diplomatic *coup*<sup>E3</sup> just before *leaving*<sup>E4</sup> Israel when Prime Minister Benjamin Netanyahu *apologized*<sup>E5</sup> to Turkey for a *2010*<sup>T2</sup> commando *raid*<sup>E6</sup> that *killed*<sup>E7</sup> nine activists on a Turkish vessel on a Gaza bound flotilla.

An interpretation of the temporal ordering of these events is:

E2 (scoring), E3 (coup) and E5 (apologized) are SIMULTANEOUS events. They occur BEFORE E4 (leaving), which occurs BEFORE E1 (arrived). E1 IS\_INCLUDED in T1 (Friday). E6 (raid) IS\_INCLUDED in T2 (2010) and E7

(killed) IS\_INCLUDED in E6. Events E1, E2, E3, E4, E5 occurred AFTER T2 (2010).

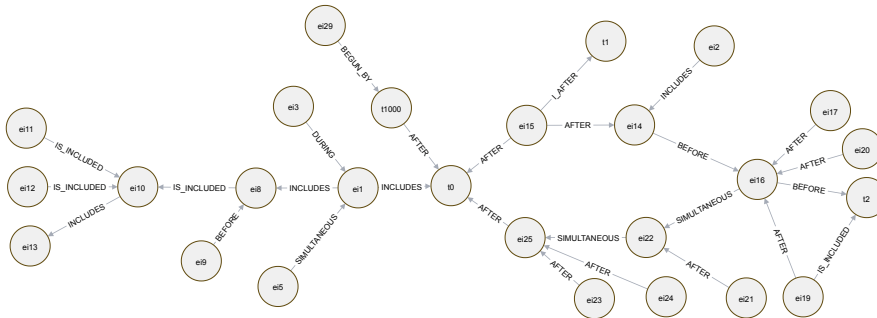
However, it is not clear if E2, E3 and E5 occurred on or before T1 (Friday).

### 2.3 Event Graphs

After a text is marked up in TimeML, Allen’s algebra can be used to assess temporal relationships. Event graphs (EG) are a useful way to represent events (Glavaš and Šnajder, 2015). An event graph is a labeled graph in which nodes represent individual events, and edges represent semantic relations between events (e.g., temporal relations). We can adapt EGs to model TimeML annotated text where nodes are the union of events and times and arcs are the TimeML tlinks, with arc weights representing the values of any of the fourteen types of relations or NONE (denoting no relation).

Figure 3 is an example of an EG of a human annotated BBC news feed. The manual annotation provides a “Platinum” standard benchmark against which classifiers can be evaluated.

Heavy *snow*<sup>E1</sup> is *causing*<sup>E2</sup> *disruption*<sup>E3</sup> to transport across the UK, with heavy *rainfall*<sup>E1000</sup> *bringing*<sup>E5</sup> *flooding*<sup>E1001</sup> to the south-west of England. Rescuers *searching*<sup>E6</sup> for a woman *trapped*<sup>E7</sup> in a landslide at her home in Looe, Cornwall, *said*<sup>E8</sup> they had *found*<sup>E9</sup> a body.



**Fig. 3** Event graph of hand annotated news feed (bbc\_20130322\_1600).

Figures 4 and 5 show the EGs produced by two participants in the news-feed challenge for the same BBC news feed. Edge weights are omitted for clarity of the illustration. The EGs demonstrate the diversity between the classifier outputs and the challenge in reconciling conflicting recommendations from the individual classifiers.

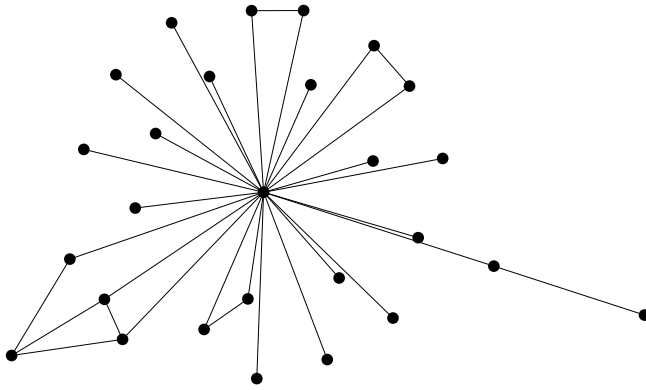


Fig. 4 EG of news feed (bbc\_20130322\_1600) annotated by Cleartk-2 (Bethard, 2013).

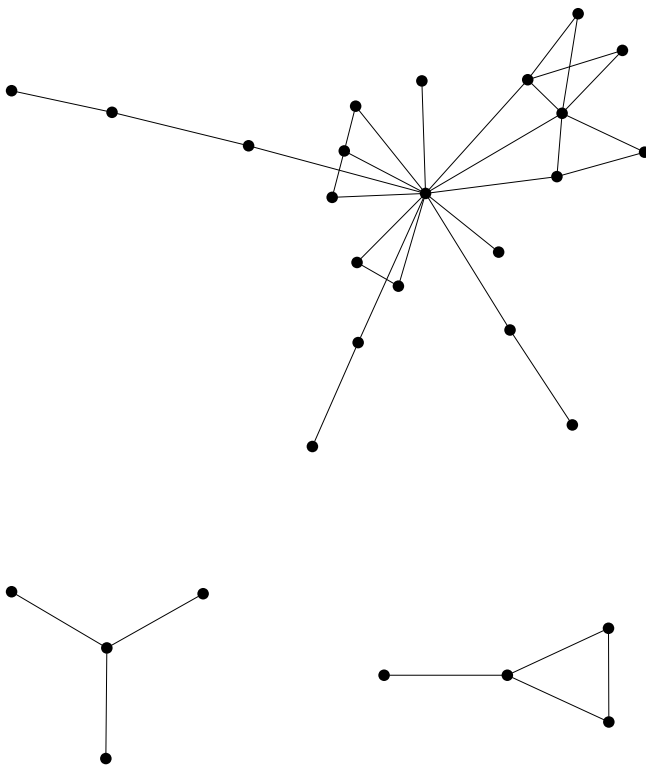


Fig. 5 EG of news feed (bbc\_20130322\_1600) annotated by Navytime-1 (Chambers, 2013).

## 2.4 Inference via Integer Programming

The focus of our study is to reconcile the reltype labelling predictions of the multiple classifiers by considering the transitive closure.

In general, any ensemble combines the results of multiple classifiers in order to improve on their individual performance (Florian et al., 2002; Dietterich, 2000). Diversity in predictions of the individual classifiers allows for an ensemble of classifiers to be more accurate than any individual classifier, but does not guarantee it. Consider, for example, the majority vote as a simple ensemble method: if there is one individual classifier with 100% accuracy, the majority vote may still result in less than 100% accuracy of the ensemble. The improvement of the accuracy depends on the numerous design choices in the construction of the ensemble, such as weights in a variant of our majority vote example. There is a good mathematical theory (Cesa-Bianchi and Lugosi, 2006, Chapter 2 and 5) for the construction of ensembles *without consistency constraints*, and correspondingly, a wealth of algorithms. Bagging and boosting are two particularly popular algorithms for learning weights to combine the multiple classifiers into an ensemble. We refer to (Trevor et al., 2009, Chapter 16) for an excellent overview or book-length surveys of Zhou (2012); Rokach (2010); Seni and Elder (2010).

Turning our attention then to additionally enforcing consistency constraints, we recall that as discussed in Section 2, computing transitive closure in Allen’s interval algebra is NP-Hard (Nebel and Bürckert, 1995). Heuristics are often used for NP-Hard problems, but may sacrifice speed for accuracy. For this study, we chose Integer Programming (IP), also known as Integer Linear Programming (ILP). IP is a constrained optimisation technique, with a long history of reach, as documented in Schrijver (2003); Nemhauser and Wolsey (1988); Papadimitriou and Steiglitz (1998), and there are many high-performing IP solvers. IP is a leading technique for NP-Hard combinatorial optimisation problems, where we search for an optimum object within a finite collection. There are naturally suggested IP formulations where a binary decision variable represents the choice being made e.g., an item is either selected, or it is not selected. In our case of reconciling reltype recommendations, we are seeking a decision on which reltype to select: a reltype is either selected, or it is not selected.

In contrast to heuristics, IP is an exact (global optimisation) technique. We can guarantee finding the optimal solution, if one exists and its objective function is bounded. For NP-Hard problems, IP solvers employ a Linear Programming (LP) relaxation with branch-bound-and-cut methods. In the LP relaxation, a linear function is optimised over an intersection of  $m$  linear inequalities and equalities (so called polyhedron) to find the optimal value for

an  $n$ -vector of decision variables  $x_j, j = 1, 2, \dots, n$ , i.e.,

$$\max \sum_{j=1}^n c_j x_j \quad (1)$$

$$\text{s.t. } \sum_{j=1}^n a_{i,j} x_j \leq b_i \quad i = 1, 2, \dots, m \quad (2)$$

$$x_j \in \mathbb{R}^+, \quad j = 1, 2, \dots, n. \quad (3)$$

where  $c_i, b_i$  and  $a_{i,j}$  are known real valued coefficients and  $\mathbb{R}^+$  denotes non-negative real numbers. Since the solution space enclosed by the set of linear inequalities is a convex polyhedron, we are guaranteed that the optimal solution occurs at a corner point of the polyhedron. LP solvers find the optimal continuous relaxation in polynomial time in the dimension  $n$ . However, constraining the variables to be integer makes the IP problem NP-Hard, and may require numerous relaxations to be solved. Despite this fact, the computational performance of modern IP solvers often turns out to be very good.

IP has been applied to a wide range of NP-Hard problems in operations research, software verification, mathematical logic, combinatorics, and recently, computational linguistics. Examples of the latter include text summarisation (Woodsend and Lapata, 2011), semantic role labelling (Roth and Yih, 2004; Punyakanok et al., 2004), global reconciliation on temporal labels (Chambers and Jurafsky, 2008), coupling of local event-event and event-time classifiers (Do et al., 2012). In these cases, the IP formulation makes it possible to enforce global constraints over subgraphs of intervals only; for example UzZaman et al. (2013) enforces global constraints over three intervals. In our study, the rules of interval algebra are applied over the *whole* test text passage.

In particular, we wish to decide the “best” arc labelling which is consistent with the given interval information for the complete passage of text by reconciling the recommendations from different classifiers. The decisions are integer in nature, i.e., we must select exactly one of the retypes for arcs as suggested in the transitive closure composition table.

### 3 The IP Ensemble Method

~~So far we have described Allen’s interval algebra, the TimeML temporal information markup language to label temporal events in texts, the use of temporal event graphs to model TimeML annotated text, and described decision making by integer programming. We now show how to construct an ensemble that uses the outputs of multiple classifiers and enforces consistency constraints.~~

Our main goal is to construct an ensemble that enforces the consistency constraints of Allen’s interval algebra. Our approach is to formulate the problem as an IP model. We evaluate the performance of the IP models empirically on NLP competition test instances. The Temporal information extraction challenge (TempEval) (Verhagen et al., 2009), and Semantic Evaluation (SemEval)

competitions provide test text instances, and specify event and temporal relation processing and labelling tasks. They use standardised data formats, TimeML markup tagging, and performance measures so that competitors systems’ performances on the tasks can be evaluated. We test our IP ensemble approach on two of the tasks: linking temporal events across news-feed documents, and in clinical notes and pathology reports.

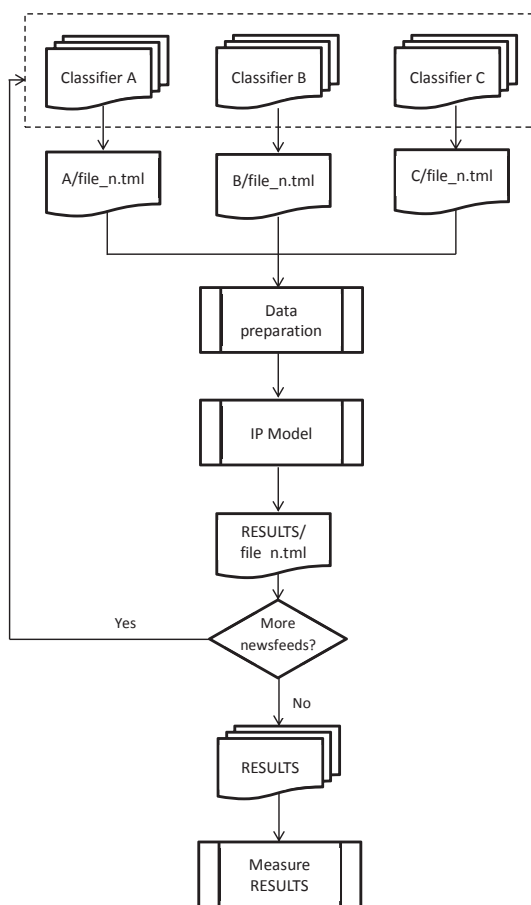
Leading participants in the news-feed and clinical-notes challenges predict the relation between any two temporal events with some success. Recall, however, the diversity of classification results observed in Figures 4 and 5: participants produce conflicting predictions. We use an IP ensemble to reconcile the diverse predictions of the individual classifiers, which has been kindly provided by the participants, by enforcing the transitivity rules of the underlying relation algebra.

The ensemble method could be seen as a pipeline, which takes the output from participant classifiers, converts the individual classifier output to input of our IP model, which reconciles conflicts between the participant classifiers, and decides a globally consistent interpretation. Figure 6 shows an overview of the data flows for a sample of three classifiers  $A, B$  and  $C$ . In particular, the conversion of the individual classifier output in TimeML format, which includes all the TLINKS, to an event graph (EG), is performed in a data preparation step. We input the EGs to an IP model. We solve the IP model using a state-of-the-art solver to obtain the most likely relation for all EG arcs, subject to global constraints. Next, we convert the IP solutions to a TimeML-formatted file. Finally, we evaluate with the TimeML-formatted file with the temporal-evaluation tool, which has been used in the competition, *temporal\_evaluation.py* from (UzZaman, 2012).

The details, including the composite relations for path consistency, depend on the [challenge: the composite relations for our ensemble are given in Tables 1 and 2](#). In the case of the clinical note challenge, the subset of temporal relations in the clinical data did not include all inverse relations. It was hence necessary to introduce some additional relations to cover all possibilities. For example, if classifier1 detects  $I_1$  CONTAINS  $I_2$  and classifier2 detects  $I_2$  CONTAINS  $I_1$ , we capture the output of classifier2 as using relation CONTAINS\_INV. Should, however, our eventual output be  $I_1$  CONTAINS\_INV  $I_2$ , in order to be understood by the evaluation program, it has to be encoded as  $I_2$  CONTAINS  $I_1$ . We also note that the inverse of OVERLAP is OVERLAP.

### 3.1 The Integer-Programming Model

To describe our IP ensemble formulation, we use  $\mathcal{A}$  to denote the set of annotated tlinks, i.e., the arcs in the transitive closure of an EG,  $R$  to denote the reltypes, and  $\mathcal{C}^*(r_1, r_2) : R \times R \rightarrow 2^R$  to denote the set of possible composite relationships of reltypes  $r_1, r_2$ . In the case of SemEval-2013 TempEval-3 Task C,  $R = \{p, pi, o, oi, I, Ii, m, mi, s, si, f, fi, NONE\}$  is the set of reltypes considered, and Table 1 captures the corresponding  $\mathcal{C}^*$ .



**Fig. 6** A sketch of the ensemble data processing pipeline

In the case of TempEval 2016 Task 12,  $R = \{p, pi, c, ci, o, b, bi, NONE\}$  and Table 2 captures the corresponding  $\mathcal{C}^*$ .

Let  $x_{i,j}$  be a binary decision variable:

$$x_{i,j} = \begin{cases} 1, & \text{if relationship } j \text{ is assigned to links } i \\ 0, & \text{otherwise.} \end{cases}$$

The integer program can be seen as a weighted assignment problem with additional constraints:

$$\max \sum_{i \in \mathcal{A}} \sum_{j \in R} \alpha_{i,j} x_{i,j} \quad (4)$$

$$\text{s.t. } \sum_{j \in R} x_{i,j} = 1, \quad \forall i \in \mathcal{A} \quad (5)$$

$$x_{lm,a} + x_{mn,b} - \sum_{j \in \mathcal{C}^*} x_{ln,j} \leq 1, \quad \forall lm, mn, ln \in \mathcal{A}, a, b \in R \quad (6)$$

$$x_{i,j} \in \{0, 1\}. \quad (7)$$

Eq. (4) is the objective, which maximises the sum of weights assigning tlinks to relation types.

Eq. (5) are mutual exclusivity constraints, which guarantee that exactly one relation type is assigned to each tlink. Eq. (6) performs a transitive composition consistency check on each triplet of events  $\{l, m, n\}$ , where tlink  $lm$  links events  $l$  and  $m$ ,  $mn$  links events  $m$  and  $n$ , and  $ln$  links events  $l$  and  $n$ . Finally, Eq. 7 are the binary integrality constraints.

We take as an example the earlier arc labelling task in Subfigure 2.c. We represent the tlinks between intervals  $(I_1, I_2)$  by arc  $lm$  in an event graph. Similarly, we represent the tlink between  $(I_2, I_3)$  by arc  $mn$ .

The reltypes of tlinks  $lm$  and  $mn$  are Contains (c) and Precedes (p). Eq. 5 says we must assign a reltypes to arc  $ln$ . Using  $R$  from TempEval 2016 Task 12 as an example, then summing over  $R$  we get:

$$x_{ln,p} + x_{ln,pi} + x_{ln,c} + x_{ln,ci} + x_{ln,o} + x_{ln,b} + x_{ln,bi} + x_{ln,NONE} = 1$$

Eq. 6 restricts the choice to one of the reltypes for arc  $ln$  from the transitive composition Table 2, so we get:

$$x_{lm,c} + x_{mn,p} - x_{ln,p} - x_{ln,c} - x_{ln,o} - x_{ln,bi} \leq 1$$

Since the decision variables are binary integers, when  $x_{lm,c} = x_{mn,p} = 1$ , Eq. 6 forces the choice of one of the four consistent reltypes in a feasible solution.

The IP solver finds binary integer solution values for the decision variables, which represent the tlinks in the ensemble’s reconciled predictions. While other ensemble techniques such as bagging learn weighted combinations of participating classifiers’ recommendations, they do not harness the knowledge of the feasible reltype composition given to us via Allen’s transitive closure. The IP inequalities encode this knowledge as an explicit set of constraints simultaneously.

Notice that the objective function weights  $\alpha_{i,j}$  are calculated in the data preparation step. This allows a wealth of weight functions to be explored. In Section 4 we describe computational experiments where  $\alpha_{i,j}$  is e.g., the sum of F1 scores of classifiers suggesting the reltype, or a convex combination of the classifiers’ precision and recall. Other options are to use probability

values based on the empirical probability distribution of assignments from the individual classifiers (Chambers and Jurafsky, 2008), but the F1 metric seems more robust, as it considers both precision and recall. We discuss our approach to selecting participating classifiers in Section 4.

### 3.2 SemEval and TempEval NLP Competitions and Evaluation Metrics

In this section we provide some detail of the news-feed and clinical-notes temporal-reasoning challenges, and how success is measured. SemEval-2013 TempEval-3 Task C participants trained their classifiers on a training data set and tested them on a platinum dataset of twenty news-feeds considered all fourteen TimeML temporal relations (UzZaman et al., 2013; UzZaman, 2012). For brevity, we refer to this as the news-feed challenge in our computational results.

Clinical TempEval was introduced in SemEval-2016. The focus of TempEval 2016 Task 12 was clinical data (i.e., medical records) and six out of the fourteen relations of Allen’s interval algebra. Participant systems were trained and evaluated on a corpus of clinical and pathology notes from the Mayo Clinic, annotated with an extension of TimeML for the clinical domain.

We are interested in Phase 2 of the competition, where participants were provided with manually annotated EVENTS and TIMEX3s, annotated by the THYME project ([thyme.healthnlp.org](http://thyme.healthnlp.org)). The participating classifiers used the same baseline data to extract temporal relations between known events and time expressions. Phase 2 was further subdivided into two parts: First, to identify relations between events and the document creation time (DOC-TIMEREL); second, to identify narrative container relations (CONTAINS). A narrative container can be thought of as a temporal bucket into which an EVENT or series of EVENTS may fall (Pustejovsky and Stubbs, 2011). These narrative containers are often represented (or “anchored”) by dates or other temporal expressions (within which a variety of different EVENTS occur). The objective is to link EVENTS to narrative containers and then link those containers so that contained EVENTS can be linked by inference. For brevity, we refer to this as the clinical-notes challenge in our computational results.

The news-feed and clinical notes challenges were similar in complexity: classifiers were required to identify relations between events/time expressions *and* to label those relations. Systems only received credit for a narrative container relation if they found both events/times and correctly assigned a CONTAINS relation between them.

For consistency, both challenges use the same evaluation tool and metrics of Precision (P), Recall (R), and a harmonic mean thereof (F1 score). To formalise these following UzZaman et al. (2013) and Bethard et al. (2016), consider the set  $S$  of items predicted by the system and the set  $H$  of items annotated by the humans.  $S_c$  and  $H_c$  are the closed sets of temporal relations, i.e., additional relations that are inferred from other relations.  $S_r$  and  $H_r$  are

the reduced sets of relations, i.e., with redundant relations removed.<sup>1</sup> Then, P, R, and F1 score are defined as:

$$P = \frac{|S_r \cup H_c|}{|S_r|} \quad R = \frac{|S_c \cup H_r|}{|H_r|} \quad F1 = \frac{2 \times P \times R}{P + R}$$

The  $F_1$  score, in particular, is regarded as useful to measure the balance between Precision (which measures how many selected items are relevant) and Recall (which measures how many relevant items are selected). It gives greater importance to low values compared to a simple average. We would ideally like to optimise both Precision and Recall, or at least understand their mutual trade off. Powers (2011) examines the relationships between these measures, and notes that none takes the number of True Negatives into account. While the nature of the underlying interval algebra in our work, which includes relations and their converses, partly compensates for the focus on P, R and  $F_1$  scores, Flach (2003) suggests the use of Receiver Operating Characteristics (ROC), a concept from Signal Processing, to give a geometric representation of the trade-off between the precision and recall.

All details of our implementation are publicly available in the form of the source code at <https://github.com/catherinekerr/ip-ensemble>, which is licensed under Apache License 2.0.

## 4 Computational Experiments

Next, we set out to evaluate the IP ensemble. Our goal is to empirically evaluate design options for the objective function coefficients, and to evaluate which classifiers should be included in the IP ensemble. We aim to answer the following questions:

- What is the statistical performance of a simple hand-picked ensemble?
- What is the statistical performance of the best possible ensemble, in terms of the subset of individual classifiers?
- What is the trade-off between precision and recall across all possible ensembles in terms of the subset of individual classifiers?
- Shall one use weights based on precision, recall, or their arithmetic mean?
- How fast can one solve the IP models in practice?

We conducted two sets of computational experiments. In the first set of experiments, we create IP ensembles for results from the news-feed challenge (Uz-Zaman et al., 2013). We present results and analysis from these experiments in Section 4.1. Results from the news-feed IP ensemble looked promising, but the amount of data is limited to only 20 news-feeds, split into training/test sets of 10 news-feeds each. In the second set of experiments, we validated the use of an IP ensemble on the larger datasets from the clinical-notes challenge (Bethard et al., 2016). We present the results from the second set of experiments in Section 4.2.

<sup>1</sup> In Clinical TempEval 2016, sets were not reduced, so  $S_r = S$  and  $H_r = H$ .

Throughout both sets of experiments, we have used an implementation of the pipeline of Figure 6. This is composed of the code to formulate IP ensembles (Equations 4–7), an IP solver, a variety of parsers, and data-conversion routines. In particular, we formulate each IP instance for the ensemble of classifiers in Pyomo (Hart et al., 2012). This allows us to switch between IP solvers seamlessly. It also allows for the execution on a variety of platforms, ranging from laptops to clusters. We use both cbc, the open-source IP solver (Forrest and Lougee-Heimer, 2005), and IBM ILOG CPLEX 12.4, the state-of-the-art commercial solver, and compare their performance on a single core of a PC equipped with an Intel Xeon E7458 processor clocked at 2.4GHz. We tested several objective function weights. For brevity we focus on results using F1 scores, which worked best, and indicate where other weights are used. We denote the competition participants’ classifiers as “individual classifiers” in the results below, to distinguish them from our IP ensemble classifiers.

#### 4.1 News-Feed IP Ensemble Results

The news-feed dataset from SemEval-2013 TempEval-3 Task C competition was split into two parts. Eleven individual classifiers were trained on the training set, and evaluated on the Platinum set. The participants kindly provided their classifications for the Platinum dataset to us. We carried out extensive computational tests, initially using human intuition, with some trial and error, to compose ensembles that improved upon previous best F1 scores. We subsequently used an automated approach to generate all IP ensembles of combinations of the 2 to 11 individual classifiers. Since the dataset is relatively small, we explored two training and testing procedures.

Table 3 illustrates how some of the leading individual classifiers, our IP ensemble and the human annotators (Platinum) annotated the events in the Obama example in Section 2.

**Table 3** Sample of classifier tlinks

	E2 (scoring) E4 (leaving)	E4 (leaving) T1 (Friday)	E2 (scoring) T1 (Friday)
Cleartk-1	BEFORE	NONE	NONE
Cleartk-2	NONE	NONE	NONE
Cleartk-3	BEFORE	NONE	NONE
Cleartk-4	BEFORE	NONE	NONE
Navytime-1	BEFORE	NONE	IS_INCLUDED
Navytime-2	SIMULTANEOUS	IS_INCLUDED	NONE
UTTime-4	SIMULTANEOUS	IS_INCLUDED	IS_INCLUDED
UTTime-5	SIMULTANEOUS	NONE	IS_INCLUDED
Ensemble	BEFORE	IS_INCLUDED	BEFORE
Platinum	BEFORE	NONE	BEFORE

Table 4 summarises the individual classifiers’ precision, recall, and F1 score on the Platinum dataset. We denote the individual classifier’s by an ID to identify which classifiers participate in our IP ensembles.

**Table 4** News-Feed Results: The performance of the individual classifiers on the Platinum dataset.

ID	Classifier	F1	Precision	Recall
C1	cleartk-1	0.3517	0.3764	0.3300
C2	cleartk-2	0.3624	0.3732	0.3521
C3	cleartk-3	0.3421	0.3336	0.3510
C4	cleartk-4	0.3594	0.3526	0.3664
N1	navytime-1	0.3079	0.3519	0.2737
N2	navytime-2	0.3588	0.5078	0.2774
U1	UT-1	0.2428	0.1490	0.6556
U2	UT-2	0.2415	0.1487	0.6424
U3	UT-3	0.2422	0.1507	0.6170
U4	UT-4	0.2882	0.3752	0.2340
U5	UT-5	0.3499	0.3605	0.3400

#### 4.1.1 News-Feed Train-Test Procedure 1

Firstly, the individual classifiers were trained on the training set and tested on the Platinum set of SemEval-2013 to obtain their F1 scores, which we used as the objective function coefficients for the IP ensembles.

Table 5 shows sample results for this approach, Procedure 1. We summarise the *training* F1 score, precision, and recall of several variant ensembles on the Platinum set of 20 news-feeds. The IP ensemble (C2, C4, U4+5, N1+2) using 6 of the 11 individual classifiers (ClearTK-2, ClearTK-4, UTTime-4, UTTime-5, and Navytime-1 and Navytime-2), results in the best performance with F1 of 0.3899 and recall of 0.5. These *training* scores are considerably better than the *test* performance of the individual classifiers, as summarised in Table 4.

**Table 5** News-Feed Results: Procedure 1 with IP ensembles composed of classifiers’ IDs as in Table 4.

IDs	F1	Precision	Recall
C1-3	0.3583	0.3515	0.3653
C1-4	0.3600	0.3488	0.3720
C2, C4, U4, N1	0.3756	0.3126	0.4702
C2, U4, N1	0.3786	0.3205	0.4625
C2, C4, U4+5, N1	0.3862	0.3182	0.4912
C1-4, U4+5, N1+2	0.3877	0.3166	0.5000
C1-2, C4, U4+5, N1+2	0.3893	0.3192	0.4989
C2, C4, U4+5, N1+2	0.3899	0.3195	0.5000

However, we do note that this comparison is not quite fair as the IP ensembles are building on the classifiers results for the Platinum instances, we are effectively re-using the same instances in training and testing. We next evaluated splitting the Platinum dataset to test our IP ensembles.

#### 4.1.2 News-Feed Train-Test Procedure 2

In the second procedure, we randomly split the Platinum dataset of SemEval-2013 into two subsets, S1 and S2, of ten news-feeds each. The individual classifiers, which were trained on the original training set, were evaluated on S1 to extract F1 scores for use in the IP ensemble objective function.

Table 7 details the *test* F1 scores, precision, and recall for a sample of manually picked IP ensembles tested on the remaining 10 news-feeds of S2, which used F1 scores as the objective function coefficients.

Table 8 shows results for an automated search of ensemble classifier combinations, with alternative objective function coefficients.

Throughout, the *test* scores on S2 are considerably better than the *test* performance of the individual classifiers on S2, as presented in Table 6. Notice that the F1 score of each individual classifier is worse on S2 than on  $S1 \cup S2$ , as per Tables 4, which may suggest why results of the ensembles are often worse on S2 than on  $S1 \cup S2$ . Overall, the results of the ensembles on S2 in Tables 7 and 8 provide convincing evidence of the benefit of the ensemble method.

**Table 6** News-Feed Results: The test performance of the individual classifiers on S2, the subset of Platinum.

ID	Classifier	F1	Precision	Recall
C1	cleartk-1	0.3140	0.3302	0.2993
C2	cleartk-2	0.3358	0.3398	0.3318
C3	cleartk-3	0.3240	0.3144	0.3341
C4	cleartk-4	0.3331	0.3320	0.3341
N1	navytime-1	0.2760	0.3145	0.2459
N2	navytime-2	0.2347	0.2899	0.1972
U4	UT-4	0.2599	0.3074	0.2251
U5	UT-5	0.3293	0.3224	0.3364

In particular, Table 7 captures one example of an iterative construction of an ensemble, driven by intuition of a human in the loop. Considering that good ensembles are composed of classifiers from each of the classifier groups, one could start with ClearTK-2, UTTime-4, and Navytime-1, which we refer to as C2, U4, and N1 in the first row of Table 4. Subsequently, further classifiers were added, as suggested by further rows of Table 4. Notice, for example, that the ensemble C2-U4-N1, which is diverse, outperforms C1-C2-C3-C4, where the classifiers provide similar results, despite the higher F1 scores of the individual classifiers.

**Table 7** News-Feed Results: Procedure 2 sample manually-picked IP ensembles results on S2, the subset of Platinum.

IDs	F1	Precision	Recall
C2, U4, N1	0.3495	0.2857	0.4501
C2+4, U4, N1	0.3586	0.2941	0.4594
C2+4, U4+5, N1+2	0.3602	0.2917	0.4710
C1-4, U4+5, N1+2	0.3643	0.2952	0.4756
C2+4, U4+5, N1	0.3649	0.2988	0.4687
C1+2, C4, U4+5, N1+2	0.3671	0.2962	0.4826

#### 4.1.3 News-Feed Results: Alternative Objective Function Coefficients

Next, Table 8 documents an automated search for the best IP ensemble. We evaluate the performance of 6105 IP ensembles, ranging from 2 to 11 individual classifiers, with weight  $w$ , a convex combination of precision and recall, used to explore objective functions in the IP ensembles:

$$wP + (1 - w)R.$$

Varying  $w$  allows us to create the IP objective functions coefficients as Precision (when  $w = 1$ ), Recall (when  $w = 0$ ), or an average (when  $w = 0.5$ ).

The precision, recall, and F1 scores in Table 8 are results for the IP ensembles tested on S2, the second subset. The F1 scores exceeding 0.39 improve on the human intuition effort to combine the individual classifiers and the individual classifiers, considerably.<sup>2</sup> We note that good ensembles can be obtained with each of the IP objective function weights: for example the classifier C1-C2-U4-U5 delivers the same F1 score of 0.3903, independent of whether one uses precision, recall, or their mean as the weights. We hence ascribe the improvement largely to the consistency constraints. This emphasises the contribution of an IP ensemble approach, compared to other ensemble techniques which rely on learning good weightings.

The automated search yields a number of further insights into the structure of the “search space”, as illustrated in Figures 7 and 8. In Figure 7, we plot results of ensembles, whose scores were not dominated by other ensembles in the sample (in the Pareto sense) in black or blue. In blue, in particular, we plot the ensembles of Table 8. For comparison, we plot the results of ensembles of Table 7 in green and the results of the individual classifiers in Table 7 in red. This suggests the shape of the receiver operating characteristic (ROC) curve<sup>3</sup>. As should be expected from classifiers with a finite number of classes, applied a finite number of times, this is not a proper continuous curve, but rather a

<sup>2</sup> We note these numbers are directly comparable to the performance of individual classifiers on S2, as captured in Tables 6 and 7, but not the performance of the individual classifiers on  $S1 \cup S2$ , as documented in Table 4.

<sup>3</sup> The ROC curve is usually plotted in terms of recall (true positive rate) and the false positive rate, rather than precision. We use recall and precision to keep the values directly comparable with Tables 4–8.

**Table 8** News-Feed Results: Procedure 3 sample results for automatically constructed IP ensembles, with alternative objectives  $w = 1.0$  weighing with precision,  $w = 0.0$  weighing with recall, and  $w = 0.5$  weighing with their arithmetic mean.

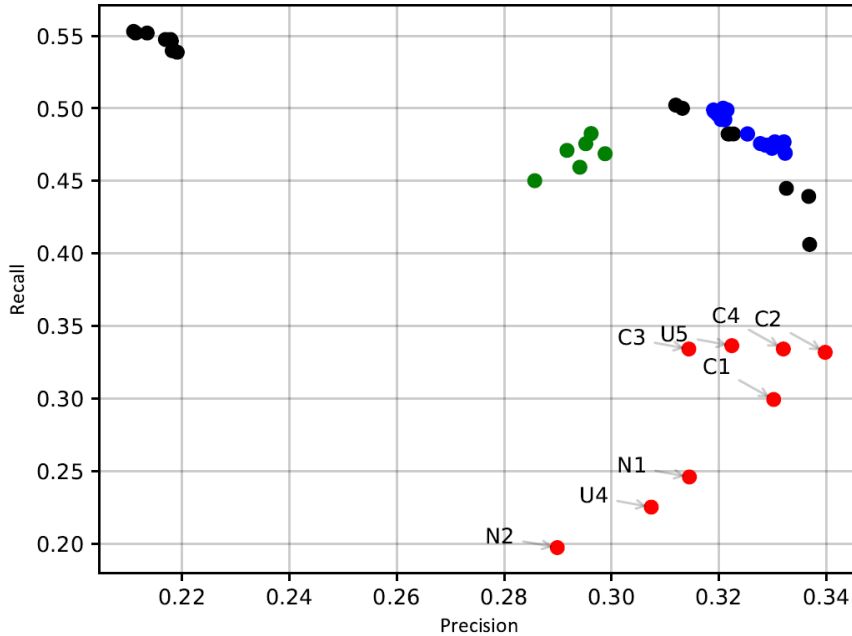
IDs	$w$	F1	Precision	Recall
C1-C2-U4-U5	0.0	0.3881	0.3277	0.4757
C1-N1-U5	0.5	0.3882	0.3204	0.4923
C1-N1-U5	1.0	0.3882	0.3204	0.4923
C2-U5	1.0	0.3884	0.3287	0.4746
C2-U5	0.5	0.3884	0.3287	0.4746
C1-C2-U5	1.0	0.3884	0.3287	0.4746
C1-C2-U5	0.5	0.3884	0.3287	0.4746
C2-U5	0.0	0.3885	0.3299	0.4724
C1-N1-U5-N2	0.0	0.3885	0.3209	0.4923
C2-U4-U5	0.0	0.3886	0.3289	0.4746
C2-C4-U4-U5	1.0	0.3886	0.3253	0.4823
C1-N1-U5	0.0	0.3886	0.3211	0.4923
C1-N1-U4-U5-N2	0.0	0.3887	0.3211	0.4923
C2-N1-U5	0.0	0.3887	0.3198	0.4956
C1-N1-U5-N2	1.0	0.3889	0.3209	0.4934
C1-N1-U5-N2	0.5	0.3889	0.3209	0.4934
C1-C2-N1-U4-U5	0.0	0.3889	0.3191	0.4978
C1-U5	0.5	0.3890	0.3323	0.4691
C1-U5	1.0	0.3890	0.3323	0.4691
C2-N1-U5-N2	0.0	0.3891	0.3199	0.4967
C1-C2-N1-U4-U5-N2	0.5	0.3892	0.3190	0.4989
C1-C2-N1-U4-U5-N2	1.0	0.3892	0.3190	0.4989
C1-N1-U4-U5-N2	1.0	0.3892	0.3209	0.4945
C1-N1-U4-U5-N2	0.5	0.3892	0.3209	0.4945
C1-C2-N1-U5	0.0	0.3903	0.3210	0.4978
C1-C2-U4-U5	1.0	0.3903	0.3304	0.4768
C1-C2-U4-U5	0.5	0.3903	0.3304	0.4768
C1-C2-N1-U4-U5-N2	0.0	0.3908	0.3208	0.5000
C1-C2-N1-U5-N2	0.0	0.3910	0.3215	0.4989
C1-C2-U5	0.0	0.3915	0.3321	0.4768

finite collection of points; indeed, in this setting, one can obtain only a finite number of outcomes, even if one varies the weights continuously. Further, the points are clustered, based on the composition of the constituent individual classifiers. In Figure 8, we also include the results of dominated ensembles (in gray and partially transparent), where the clustering is yet more pronounced.

## 4.2 Clinical-Notes Results

Our second set of experiments used TempEval 2016 Task 12. There, the clinical data was composed of 151 files, 100 of which were used in the narrative container relations part of the challenge. We split the data from each participating classifier and from the “gold”, human-annotated data into ten randomly sampled  $A/B$  datasets of 50 files each, where  $A_n = \overline{B}_n$  and  $n \in \{0, 1, \dots, 9\}$ . The dataset  $A_n$  from each classifier (and from the “gold” set) contained the same set of files; likewise for each dataset  $B_n$ . We obtained output data from

**Fig. 7** SemEval-2013 TempEval-3 Task C results: Procedure 2 results for a sample of ensembles, with each ensemble’s results represented by one data point in the precision-recall plane. In black or blue, we plot the results of ensembles, which were not dominated by other ensembles in the Pareto sense. In blue, in particular, we plot the ensembles of Table 8. In green, we plot the results of ensembles of Table 7 for comparison. In red, we plot the results of the individual classifiers, described by IDs introduced in Table 6, for comparison.

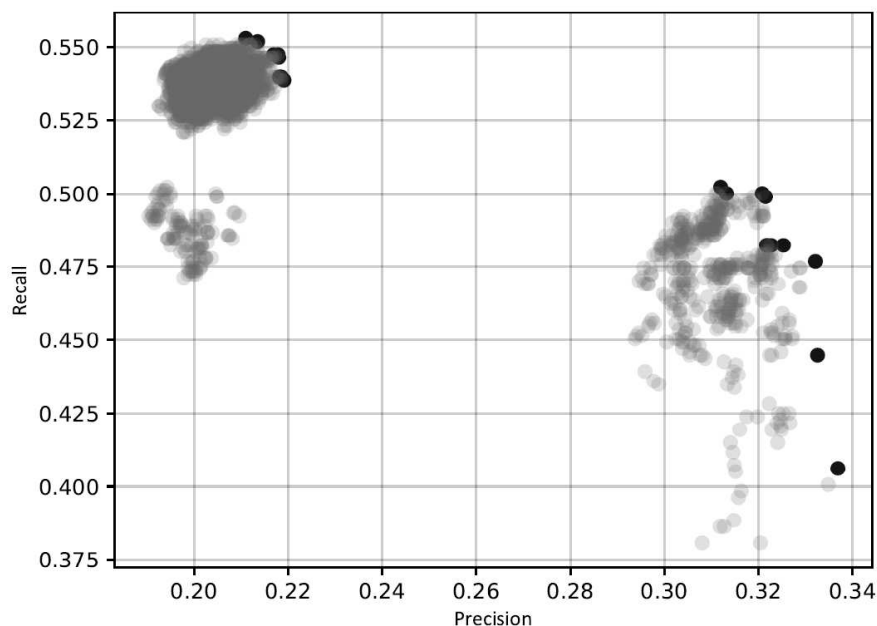


the seven leading participants in this task, comprising of the output of 13 distinct classifiers<sup>4</sup>. Each classifier dataset  $A_n$  (or  $B_n$ ) was evaluated against the corresponding “gold”  $A_n$  (or  $B_n$ ) set using the same evaluation program that was used in SemEval-2016. The F1 score of each  $A_n$  set was used as the weight in ensembles built from the corresponding  $B_n$  set, and vice versa. Our testing of ensembles was carried out against all twenty datasets and results were compared against the corresponding twenty “gold” sets.

Table 9 illustrates the scores of the individual classifiers against the hand-annotated set for the full set and also the average scores for the ten A sets and ten B sets. The classifier *uthealth-p2s1* (Lee et al., 2016) achieved the highest F1 score of 0.573. This classifier also scored the highest in each of the twenty randomly sampled A/B datasets. We used the F1 scores of each of the twenty datasets for this classifier as the target to improve upon with our IP ensemble. We used the same evaluation tool *anafora.evaluation.py* that was used in the challenge. The best-known scores are highlighted in bold.<sup>12</sup>

<sup>4</sup> UtahBMI submitted corrected classifiers to the Task 12 challenge, whose results were too late for formal inclusion. We opted to use this data instead.

**Fig. 8** SemEval-2013 TempEval-3 Task C results: Procedure 2 results for a sample of ensembles, with each ensemble represented by one data point in the plot. In black, we plot results of ensembles, whose scores were not dominated by other ensembles. In gray and partially transparent, we plot results of other ensembles.



**Table 9** Clinical-Notes Results for the individual classifiers: Overall and average A/B sample results.

ID	Classifier	Precision	Recall	F1	F1	F1
		Overall	Overall	Overall	A set	B set
CC	CDE-IIITH-crf	0.493	0.185	0.269	0.271	0.268
CD	CDE-IIITH-deepnl	0.348	0.284	0.313	0.316	0.31
G	GUIR-Phase2-Run1	0.546	0.471	0.506	0.507	0.504
K1	KULeuven-LIIR-run1	0.714	0.428	0.536	0.538	0.534
K2	KULeuven-LIIR-run2	0.715	0.429	0.536	0.538	0.534
L1	LIMSI_COT-RUN1	0.704	0.436	0.538	0.54	0.537
L2	LIMSI_COT-RUN2	<b>0.751</b>	0.32	0.449	0.455	0.444
Ub1	UtahBMI-RUN1-corrected	0.711	0.372	0.489	0.489	0.489
Ub2	UtahBMI-RUN2-corrected	0.693	0.425	0.527	0.528	0.525
Ut1	uthealth-p2s1	0.588	<b>0.559</b>	<b>0.573</b>	<b>0.573</b>	<b>0.572</b>
Ut2	uthealth-p2s2	0.568	0.564	0.566	0.566	0.565
V1	VUACLTL-run1-phase2	0.642	0.345	0.449	0.451	0.448
V2	VUACLTL-run2-phase2	0.589	0.368	0.453	0.454	0.452

#### 4.2.1 Construction of the Clinical-Notes IP Ensemble

We built our IP ensembles in a piece-wise fashion, again using F1 score as the objective function coefficient. We took measurements at each stage to determine whether the results were an improvement on previous ensembles. As a starting point, we selected the three highest-scoring individual classifiers, from three different participants uthealth-p2s1, LIMSI\_COT-RUN1 (Grouin and Moriceau, 2016) and KULeuven-LIIR-run1 (Leeuwenberg and Moens, 2016). Subsequently, we added classifiers, one at a time. As expected, recall continued to improve, but precision deteriorated, due to the introduction of irrelevant links. At some point, improved recall is outweighed by poor precision and the F1 score deteriorates.

We addressed this issue by filtering the classifiers according to a precision score threshold. Classifiers above the threshold could nominate new relations to the ensemble *and* vote on the label. Classifiers below the threshold could only vote. This approach yielded further improvements in the F1 score of the ensemble. The best score was obtained using uthealth-p2s1, LIMSI\_COT-RUN1, KULeuven-LIIR-run1, GUIR-Phase2-Run1 (Cohan et al., 2016) and UtahBMI-RUN1-corrected (Khalifa et al., 2016). Interestingly, we achieved a good result (0.592) when we used a seven-classifier IP ensemble, adding CDE-IIITH-crf (Chikka, 2016) and VUACLTL-run1-phase2 (Caselli and Morante, 2016), once again supporting the assertion that diversity matters.

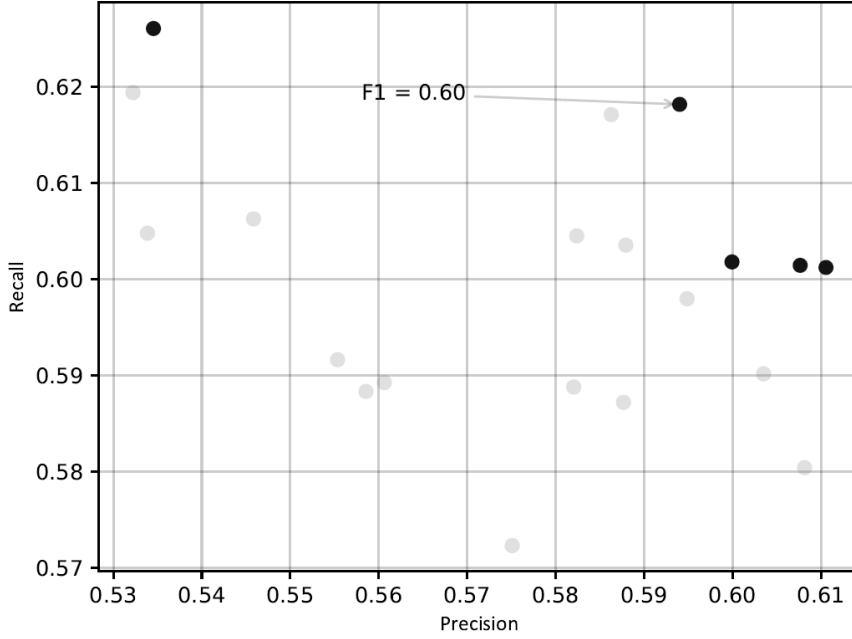
Let us illustrate the IP ensemble construction procedure in Table 10. There results are listed in the order, in which the ensembles were tested. The symbol  $\Delta$  indicates that the F1 score is better than any previous F1 scores. At each stage, the results were checked to see if the ensemble score was improving. If there was a deterioration, we increased the precision threshold, or replaced a classifier with an alternative one. Results indicate that a precision threshold of 0.65 works best. If we increased the threshold to 0.70, the recall score suffered, because too few classifiers were able to nominate relations to the ensemble. Further work might be undertaken to establish the optimum precision threshold, as it depends on the precision of the classifiers available to the ensemble and other factors. Figure 9 illustrates that this procedure actually performs rather well, in comparison with sampling the individual classifiers randomly.

#### 4.2.2 Analysis of Effect Size

Table 11 presents another analysis of the F1 score of the Clinical-Notes IP ensembles. We used Cohen’s  $d$  measure to evaluate the *effect size*, or the strength of the improvement. Cohen’s  $d$  is the standardised difference between two means:

$$d = \frac{|\bar{x}_1 - \bar{x}_2|}{s_p}$$

**Fig. 9** Results of selected Clinical-Notes IP ensembles. In black, we plot results of ensembles, whose scores were not dominated by other ensembles within the sample. In gray and partially transparent, we plot results of other ensembles.



where  $s_p$  is the pooled standard deviation of the samples of sizes  $n_1, n_2$ , respectively, with standard deviations  $s_1, s_2$ , respectively, is:

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$$

A Cohen’s  $d$  value greater than  $|1.2|$  is described as “very large”, while greater than  $|2.0|$  is described as “huge” (Sawilowsky, 2009). Table 11 shows the effect of the ensembles considering either Precision or Recall is very large, while the ensemble considering the F1 score is huge.

#### 4.3 A Remark on IP Run-time

Last but not least, let us remark on the run-time of the IP ensembles. Using CPLEX, we were able to solve even the largest IP ensemble instances in seconds. Table 12 shows the run-time performance of the IP ensemble that reconciles all 11 newsfeed individual classifiers (see Table 4) on each of the 20 newsfeeds in the challenge. This should not be surprising, considering the

**Table 10** Clinical-Notes Results: sample IP ensembles and precision threshold values

Ensemble composition	Precision threshold	Precision	Recall	F1 score	
Uthealth-p2s1	0.00	0.588	0.559	0.573	
Ut1/L1/K1	0.00	0.547	0.633	0.587	△
Ut1/L1/K1/Ub2	0.00	0.532	0.646	0.583	
Ut1/L1/K1/Ub2	0.60	0.575	0.606	0.588	△
Ut1/L1/K1/Ub1	0.60	0.586	0.597	0.589	△
Ut1/L1/G/K1/Ub1	0.60	0.587	0.599	0.590	△
Ut1/L1/G/K1/V1/Ub1	0.60	0.553	0.613	0.579	
Ut1/L1/G/K1/V1/Ub1	0.65	0.605	0.580	0.592	△
Ut1/L1/G/K1/V1/Ub1	0.70	0.636	0.533	0.575	
Ut1/L1/K1/V1/Ub1	0.65	0.605	0.578	0.591	
Ut1/L1/CC/G/K1/V1/Ub1	0.65	0.605	0.581	0.592	=
Ut1/L1/K1/Ub1	0.65	0.615	0.574	0.594	△
Ut1/L1/G/K1/Ub1	0.65	0.616	0.576	0.595	△

**Table 11** Clinical-Notes Results: IP ensembles composed of classifiers Ut1/L1/G/K1/Ub1.

Testset	Classifier Precision	Ensemble Precision	Classifier Recall	Ensemble Recall	Classifier F1 score	Ensemble F1 score
00-A	0.572	0.615	0.549	0.563	0.561	0.588
01-A	0.603	0.630	0.543	0.569	0.572	0.598
02-A	0.604	0.636	0.570	0.578	0.586	0.606
03-A	0.588	0.608	0.553	0.573	0.570	0.590
04-A	0.566	0.601	0.542	0.570	0.554	0.585
05-A	0.597	0.636	0.560	0.581	0.578	0.607
06-A	0.576	0.595	0.576	0.594	0.576	0.594
07-A	0.604	0.623	0.569	0.598	0.586	0.610
08-A	0.573	0.619	0.572	0.595	0.573	0.607
09-A	0.603	0.619	0.547	0.565	0.574	0.591
00-B	0.604	0.617	0.568	0.589	0.585	0.603
01-B	0.569	0.598	0.579	0.585	0.574	0.592
02-B	0.569	0.593	0.545	0.573	0.556	0.583
03-B	0.588	0.625	0.564	0.579	0.576	0.601
04-B	0.606	0.630	0.573	0.581	0.589	0.604
05-B	0.578	0.597	0.557	0.569	0.567	0.583
06-B	0.599	0.638	0.543	0.559	0.570	0.596
07-B	0.571	0.608	0.547	0.552	0.559	0.579
08-B	0.601	0.613	0.547	0.559	0.573	0.585
09-B	0.573	0.614	0.571	0.586	0.572	0.600
Mean	0.587	0.616	0.559	0.576	0.573	0.595
Std. dev.	0.015	0.014	0.013	0.013	0.010	0.009
Cohen's <i>d</i>		2.00		1.31		2.34

largest instance (WSJ\_20130322\_159) for the ensemble of all 11 individual classifiers had dimension of the decision variable as low as 7365. The IP ensemble instances for the Clinical-Notes challenge were solved faster still, due to the sparsity of the constraint matrix. This suggests the scalability of our IP ensemble approach to related temporal reasoning problems.

**Table 12** IP ensemble performance.

Instance	Dim.	cbc (secs)	CPLEX (secs)
AP_20130322	2,790	8.01	0.57
bbc_20130322_1150	4,155	23.94	1.02
bbc_20130322_1353	4,440	119.1	3.73
bbc_20130322_1600	2,535	16.37	1.49
bbc_20130322_332	3,675	8.56	1.01
bbc_20130322_721	2,115	11.16	0.69
CNN_20130322_1003	7,200	181.49	4.69
CNN_20130322_1243	900	0.9	0.14
CNN_20130322_248	2,025	2	0.71
CNN_20130322_314	3,045	19.86	1.19
CNN_20130322_821	285	0.04	0.01
nyt_20130321_cyprus	6,285	82.55	4.11
nyt_20130321_china	4,755	55.61	3.76
nyt_20130321_sarkozy	2,130	11.16	2.43
nyt_20130321_women	4,950	41.63	4.04
nyt_20130321_strange	2,820	34.82	1.89
WSJ_20130318_731	2,025	5.76	0.43
WSJ_20130321_1145	1,965	6.58	0.64
WSJ_20130322_804	3,270	81.87	2.56
WSJ_20130322_159	7,365	n/a	75.57

## 5 Conclusions

Building an ensemble of classifiers can provide a significant improvement in precision and recall over any individual classifiers. The improvement in recall is understandable, considering the ensemble uses the *union* of classifiers' results. Building an IP ensemble of classifiers allows not only union of individual classifiers, but also the exploration of alternative objective function weights, and the exploitation of domain specific knowledge. We have demonstrated the application of an IP ensemble for temporal reasoning by encoding rules from Allen's Interval algebra as the IP ensemble constraints.

We have demonstrated that enforcing consistency constraints over the results of even a small number of individual classifiers improves precision over the individual classifiers in two recent competitions in classifying temporal relations. This overall consistency is required in order to make practical use of the inferred temporal relations between events in real-world applications. While one cannot hope for the processing of web-scale texts using the present IP solvers in milliseconds, the approach seems well-suited to the scale of the data sets used in the competitions, which in turn may be representative of real-world datasets.

Promising directions for future research include exploring alternative weights, alternative means of combining the individual classifiers, cf. Kittler et al. (1998), and the use of soft constraints, cf. Burke et al. (2012). While this research concentrated on event and temporal relations detection and classification, the IP ensemble framework can readily be extended to other application

domains or other types of classifiers, such as for spatial expressions, where a similar algebra can be developed in terms of geo-coordinates.

*Acknowledgements* The authors would like to thank Chambers (2013); Bethard (2013); Laokulrat et al. (2013); Leeuwenberg and Moens (2016); Caselli and Morante (2016); Chikka (2016); Grouin and Moriceau (2016); Khalifa et al. (2016); Cohan et al. (2016), and Lee et al. (2016), whose work and data kindly shared with us made this research possible. Jakub Marecek has received funding from the European Union Horizon 2020 Programme (Horizon2020/2014-2020), under grant agreement no. 688380.

## References

- James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
- Naveen Ashish, Ronald Eguchi, Rajesh Hegde, Charles Huyck, Dmitri Kalashnikov, Sharad Mehrotra, Padhraic Smyth, and Nalini Venkatasubramanian. Situational awareness technologies for disaster response. In *Terrorism Informatics*, pages 517–544. Springer, 2008.
- Jan R. Benetka, Krisztian Balog, and Kjetil Nørvåg. Towards building a knowledge base of monetary transactions from a news collection. In *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries, JCDL '17*, pages 209–218, Piscataway, NJ, USA, 2017. IEEE Press. ISBN 978-1-5386-3861-3.
- Steven Bethard. Cleartk-timeml: A minimalist approach to tempeval 2013. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM)*, volume 2, pages 10–14, 2013.
- Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. Semeval-2016 task 12: Clinical tempeval. *Proceedings of SemEval*, pages 1052–1062, 2016.
- Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007. ISSN 1556-4681.
- Eric A Bier, Stuart K Card, and John W Bodnar. Entity-based collaboration tools for intelligence analysis. In *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on*, pages 99–106. IEEE, 2008.
- Edmund K Burke, Jakub Mareček, Andrew J Parkes, and Hana Rudová. A branch-and-cut procedure for the udine course timetabling problem. *Annals of Operations Research*, 194(1):71–87, 2012.
- Tommaso Caselli and Roser Morante. Vuacntl at semeval 2016 task 12: A crf pipeline to clinical tempeval. *Proceedings of SemEval*, pages 1241–1247, 2016.
- Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

- Nathanael Chambers. Navytime: Event and time ordering from raw text. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 2*, pages 73–77. Association for Computational Linguistics, 2013.
- Nathanael Chambers and Daniel Jurafsky. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 698–706. Association for Computational Linguistics, 2008.
- Veera Raghavendra Chikka. Cde-iiith at semeval-2016 task 12: Extraction of temporal information from clinical documents using machine learning techniques. *Proceedings of SemEval*, pages 1237–1240, 2016.
- Arman Cohan, Kevin Meurer, and Nazli Goharian. Guir at semeval-2016 task 12: Temporal information processing for clinical narratives. *Proceedings of SemEval*, pages 1248–1255, 2016.
- Shaun Comfort, Sujana Perera, Zoe Hudson, Darren Dorrell, Shawman Meireis, Meenakshi Nagarajan, Cartic Ramakrishnan, and Jennifer Fine. Sorting through the safety data haystack: Using machine learning to identify individual case safety reports in social-digital media. *Drug Safety*, 41(6): 579–590, Jun 2018. ISSN 1179-1942. doi: 10.1007/s40264-018-0641-7.
- Jacqueline W. Daykin, Mirka Miller, and Joe Ryan. Trends in temporal reasoning: Constraints, graphs and posets. In Ilias S. Kotsireas, Siegfried M. Rump, and Chee K. Yap, editors, *Mathematical Aspects of Computer and Information Sciences*, pages 290–304, Cham, 2016. Springer International Publishing. ISBN 978-3-319-32859-1.
- Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.
- Quang Xuan Do, Wei Lu, and Dan Roth. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687. Association for Computational Linguistics, 2012.
- Peter L Elkin, David A Froehling, Dietlind L Wahner-Roedler, Steven H Brown, and Kent R Bailey. Comparison of natural language processing bio-surveillance methods for identifying influenza from encounter notes. *Annals of Internal Medicine*, 156(1.Part.1):11–18, 2012.
- Peter A Flach. The geometry of roc space: understanding machine learning metrics through roc isometrics. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 194–201, 2003.
- Radu Florian, Silviu Cucerzan, Charles Schaefer, and David Yarowsky. Combining classifiers for word sense disambiguation. *Natural Language Engineering*, 8:327–341, 12 2002.
- John Forrest and Robin Lougee-Heimer. Cbc user guide. In *Tutorials in Operations Research*, pages 257–277. INFORMS, 2005.
- Goran Glavaš and Jan Šnajder. Construction and evaluation of event graphs. *Natural Language Engineering*, 21:607–652, 2015.
- Cyril Grouin and Véronique Moriceau. Limsi at semeval-2016 task 12: machine-learning and temporal information to identify clinical events and

- time expressions. *Proceedings of SemEval*, pages 1225–1230, 2016.
- William E Hart, Carl Laird, Jean-Paul Watson, and David L Woodruff. *Pyomo – Optimization modeling in python*. Springer Science & Business Media, 2012.
- Chung-Chi Huang and Zhiyong Lu. Community challenges in biomedical text mining over 10 years: success, failure and the future. *Briefings in Bioinformatics*, 17(1):132–144, 2016. doi: 10.1093/bib/bbv024. URL +<http://dx.doi.org/10.1093/bib/bbv024>.
- Abdulrahman Khalifa, Sumithra Velupillai, and Stephane Meystre. Utahbmi at semeval-2016 task 12: Extracting temporal information from clinical text. *Proceedings of SemEval*, pages 1256–1262, 2016.
- Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
- Peter B Ladkin. *Constraint reasoning with intervals: a tutorial, survey and bibliography*. International Computer Science Institute, 1990.
- Natsuda Laokulrat, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. Uttime: Temporal relation classification using deep syntactic features. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM)*, volume 2, pages 88–92, 2013.
- Hee-Jin Lee, Yaoyun Zhang, Jun Xu, Sungrim Moon, Jingqi Wang, Yonghui Wu, and Hua Xu. Uthealth at semeval-2016 task 12: an end-to-end system for temporal information extraction from clinical notes. *Proceedings of SemEval*, pages 1292–1297, 2016.
- Artuur Leeuwenberg and Marie-Francine Moens. Kuleuven-liir at semeval 2016 task 12: Detecting narrative containment in clinical records. *Proceedings of SemEval*, pages 1280–1285, 2016.
- Jayant Madhavan, Shawn R. Jeffery, Shirley Cohen, Xin (Luna) Dong, David Ko, Cong Yu, and Alon Halevy. Web-scale data integration: You can only afford to pay as you go. In *CIDR*, 2007.
- Dana Movshovitz-Attias, Steven Euijong Whang, Natalya Noy, and Alon Halevy. Discovering subsumption relationships for web-based ontologies. In *Proceedings of the 18th International Workshop on Web and Databases, WebDB’15*, pages 62–69, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-3627-7.
- Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about temporal relations: a maximal tractable subclass of Allen’s interval algebra. *Journal of the ACM (JACM)*, 42(1):43–66, 1995.
- George L. Nemhauser and Laurence A. Wolsey. *Integer and combinatorial optimization*. Wiley, New York;Chichester;, 1988. ISBN 9780471828198;047182819X;.
- W. Nuij, V. Milea, F. Hogenboom, F. Frasinca, and U. Kaymak. An automated framework for incorporating news into stock trading strategies. *IEEE Transactions on Knowledge & Data Engineering*, 08 2013. ISSN 1041-4347. doi: 10.1109/TKDE.2013.133.

- Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.
- David M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.
- Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1346. Association for Computational Linguistics, 2004.
- James Pustejovsky and Amber Stubbs. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160. Association for Computational Linguistics, 2011.
- James Pustejovsky, Bob Ingria, Roser Sauri, Jose Castano, Jessica Littman, Rob Gaizauskas, Andrea Setzer, Graham Katz, and Inderjeet Mani. The specification language timeml. *The language of time: A reader*, pages 545–557, 2005.
- Lior Rokach. *Pattern classification using ensemble methods*, volume 75. World Scientific, 2010.
- Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL-2004*, 2004.
- Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. Evita: A robust event recognizer for qa systems. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 700–707, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
- Roser Saurí, Lotus Goldberg, Marc Verhagen, and James Pustejovsky. Annotating events in English. TimeML annotation guidelines, 2009. Brandeis University. Version TempEval-2010.
- Shlomo S Sawilowsky. New effect size rules of thumb. *Journal of Modern Applied Statistical Methods*, 8:597–599, 2009.
- Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.
- Giovanni Seni and John F Elder. Ensemble methods in data mining: improving accuracy through combining predictions. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–126, 2010.
- William F Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154, 2014.
- Nicholas P Tatonetti, P Ye Patrick, Roxana Daneshjou, and Russ B Altman. Data-driven prediction of drug effects and interactions. *Science translational medicine*, 4(125):125ra31–125ra31, 2012.
- Hastie Trevor, Tibshirani Robert, and Friedman JH. The elements of statistical learning: data mining, inference, and prediction, 2009. 2nd edition.
- Naushad UzZaman. *Interpreting the Temporal Aspects of Language*. Thesis, University of Rochester, 2012.

- Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (\*SEM)*, pages 1–9. Association for Computational Linguistics, 2013. Also see preprint arXiv:1206.5333.
- Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Jessica Moszkowicz, and James Pustejovsky. The tempeval challenge: Identifying temporal relations in text. *Language Resources and Evaluation*, 43(2): 161–179, 2009. ISSN 1574020X, 15728412. URL <http://www.jstor.org/stable/27743609>.
- Kristian Woodsend and Mirella Lapata. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420. Association for Computational Linguistics, 2011.
- Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.