Temporal Relation Classification Based on Temporal Reasoning

Francisco Costa
University of Lisbon
fcosta@di.fc.ul.pt

António Branco
University of Lisbon
Antonio.Branco@di.fc.ul.pt

Abstract

The area of temporal information extraction has recently focused on temporal relation classification. This task is about classifying the temporal relation (precedence, overlap, etc.) holding between two given entities (events, dates or times) mentioned in a text. This interest has largely been driven by the two recent TempEval competitions.

Even though logical constraints on the structure of possible sets of temporal relations are obvious, this sort of information deserves more exploration in the context of temporal relation classification. In this paper, we show that logical inference can be used to improve—sometimes dramatically—existing machine learned classifiers for the problem of temporal relation classification.

1 Introduction

Recent years have seen renewed interest in extracting temporal information from text. Evaluation campaigns like the two TempEval challenges (Verhagen et al., 2010) have brought an increased interest to this topic. The two TempEval challenges focused on ordering the events and the dates and times mentioned in text. Since then, temporal processing has expanded beyond the problems presented in TempEval, like for instance the work of Pan et al. (2011), which is about learning event durations.

Temporal information processing is important and related to a number of applications, including event co-reference resolution (Bejan and Harabagiu, 2010), question answering (Ahn et al., 2006; Saquete et al., 2004; Tao et al., 2010) and information extraction (Ling and Weld, 2010). Another application is learning narrative event chains or scripts (Chambers and Jurafsky, 2008b; Regneri et al., 2010), which are "sequences of events that describe some stereotypical human activity" (i.e. eating at a restaurant involves looking at the menu, then ordering food, etc.).

This paper focuses on assessing the impact of temporal reasoning on the problem of temporal information extraction. We will show that simple classifiers trained for the TempEval tasks can be improved by extending their feature set with features that can be computed with automated reasoning.

2 Temporal Information Processing

The two TempEval challenges made available annotated data sets for the training and evaluation of temporal information systems. Figure 1 shows a sample of these annotations, taken from the English data used in the first TempEval. The annotation scheme is called TimeML (Pustejovsky et al., 2003).

Temporal expressions are enclosed in `TIMEX3` tags. A normalized representation of the time point or interval denoted by time expressions is encoded in the `value` attribute of `TIMEX3` elements.

Event terms are annotated with `EVENT` tags. The annotations in Figure 1 are simplified and do not show all attributes of TimeML elements. For instance, the complete annotation for the term *created* in that figure is: `<EVENT eid="e1" class="OCCURRENCE" stem="create" aspect= "NONE" tense="PAST" polarity="POS" pos="VERB">`*created*`</EVENT>`.

Several attributes describe lexical and morpho-syntactic features of these terms, such as `stem` (its dictionary form), `pos` (its part-of-speech), `tense` (its grammatical tense, if it is a verb), `aspect` (its grammatical aspect), `polarity` (whether it occurs in a positive or negative context). The `class`

```
<TIMEX3 tid="t190" type="TIME" value="1998-02-06T22:19:00"
functionInDocument="CREATION_TIME">02/06/1998 22:19:00</TIMEX3>
<s>WASHINGTON _ The economy <EVENT eid="e1">created</EVENT> jobs at a surprisingly robust pace in
<TIMEX3 tid="t191" type="DATE" value="1998-01">January</TIMEX3>, the government <EVENT
eid="e4">reported</EVENT> on <TIMEX3 tid="t193" type="DATE"
value="1998-02-06">Friday</TIMEX3>, evidence that America's economic stamina has <EVENT
eid="e6">withstood</EVENT> any <EVENT eid="e7">disruptions</EVENT> <EVENT
eid="e224">caused</EVENT> so far by the financial <EVENT eid="e228">tumult</EVENT> in Asia.</s>
<TLINK lid="l1" relType="OVERLAP" eventID="e4" relatedToTime="t193" task="A"/>
<TLINK lid="l2" relType="AFTER" eventID="e4" relatedToTime="t191" task="A"/>
<TLINK lid="l26" relType="BEFORE" eventID="e4" relatedToTime="t190" task="B"/>
```

Figure 1: Example of the TempEval annotations (simplified) for the fragment: *WASHINGTON _ The economy created jobs at a surprisingly robust pace in January, the government reported on Friday, evidence that America's economic stamina has withstood any disruptions caused so far by the financial tumult in Asia.*

attribute includes some information about aspectual type, in the spirit of Vendler (1967)—it distinguishes states from non-stative situations—, and whether the term introduces an intensional context, among other distinctions. One time expression is especially important. This is the one denoting the document's creation time (DCT) and it is annotated with the value `CREATION_TIME` for the attribute `functionInDocument`.

Temporal relations are represented with `TLINK` elements. In the TempEval data, the first argument of the relation is always an event and is given by the attribute `eventID`. The second argument can be another event or the denotation of a time expression, and it is annotated in a `relatedToEvent` or `relatedToTime` attribute in `TLINK` elements. The attribute `relType` describes the type of temporal relation holding between these two ordered entities: `BEFORE`, `AFTER` or `OVERLAP`.[1]

The TempEval challenges consider three kinds of temporal relations.[2] These correspond to the three tasks of TempEval, whose goal was to correctly assign the relation type to already identified temporal relations. Task A considers temporal relations holding between an event and a time mentioned in the same sentence, regardless of whether they are syntactically related or not. Task B considers temporal relations holding between the main event of sentences and the DCT. Finally, task C focuses on temporal relations between the main events of two consecutive sentences.

The systems participating in TempEval had to guess the relation type of temporal relations (the value of the feature `relType` of `TLINK`s), but all other annotations were given and could be used as features for classifiers. The second TempEval included additional tasks whose goal was to obtain also these remaining annotations from raw text.

The best results for the two TempEval competitions are indicative of the state-of-the-art of temporal information processing. For task A, the best participating system correctly classified 62% of the held-out test relations. For task B this was 80% and, for task C, 55%. The best results of the second TempEval show some improvement (65%, 81% and 58% respectively), but the first task was slightly different and arguably easier (only pairs of event terms of temporal expressions that are syntactically related were considered).

In this paper, we will also be working with these three types of temporal relations and dealing with similar data. Our purpose is to check whether existing solutions to the TempEval problems can be improved with the help of a temporal reasoning component.

[1] There are also the disjunctive types `BEFORE-OR-OVERLAP`, `OVERLAP-OR-AFTER` and `VAGUE`. Because they were used only for those cases where the human annotators could not agree, they are quite rare, to the point where machine learned classifiers are seldom or never able to learn to assign these values.

[2] The second TempEval considers a fourth type, which we ignore here.

2.1 Temporal Relation Classification and Reasoning

The problem of temporally ordering events and times is constrained by the logical properties of temporal relations, e.g. temporal precedence is a strict partial order. Therefore, it is natural to incorporate logical information in the solutions to the problem of ordering events and time intervals. Perhaps surprisingly, little work has explored this idea.

Our working hypothesis is that classifier features that explore the logical properties of temporal relations can be used effectively to improve machine learned classifiers for the temporal information tasks of TempEval.

The motivation for using logical information as a means to help solving this problem can be illustrated with an example from Figure 1.

There, we can see that the date `1998-02-06`, denoted by the expression *Friday*, includes the document's creation time, which is `1998-02-06T22:19:00`. We know this from comparing the normalized value of these two expressions, annotated with the `value` attribute of `TIMEX3` elements. From the annotated temporal relation with the id `l26` (the last one in the figure) we also know that the event identified with `e4`, denoted by the form *reported*, precedes the document's creation time.

From these two facts one can conclude that this event either precedes the time denoted by *Friday* or they overlap; this time cannot however precede this event. That is, the possible relation type for the relation represented with the `TLINK` named `l1` is constrained—it cannot be `AFTER`.

What this means is that, in this example, solving task B can, at least partially, solve task A. The information obtained by solving task B can be utilized in order to improve the solutions for task A.

3 Related Work

The literature on automated temporal reasoning includes important pieces of work such as Allen (1984); Vilain et al. (1990); Freksa (1992). A lot of the work in this area has focused on finding efficient methods to compute temporal inferences.

Katz and Arosio (2001) used a temporal reasoning system to compare the temporal annotations of two annotators. In a similar spirit, Setzer and Gaizauskas (2001) first compute the deductive closure of annotated temporal relations so that they can then assess annotator agreement with standard precision and recall measures.

Verhagen (2005) uses temporal closure as a means to aid TimeML annotation, that is as part of a *mixed-initiative* approach to annotation. He reports that closing a set of manually annotated temporal relations more than quadruples the number of temporal relations in TimeBank (Pustejovsky et al., 2003), a corpus that is the source of the data used for the TempEval challenges.

Mani et al. (2006) use temporal reasoning as an oversampling method to increase the amount of training data. Even though this is an interesting idea, the authors recognized in subsequent work that there were methodological problems in this work which invalidate the results (Mani et al., 2007).

Since the advent of TimeBank and the TempEval challenges, machine learning methods have become dominant to solve the problem of temporally ordering entities mentioned in text. One major limitation of machine learning methods is that they are typically used to classify temporal relations in isolation, and therefore it is not guaranteed that the resulting ordering is globally consistent. Yoshikawa et al. (2009) and Ling and Weld (2010) overcome this limitation using Markov logic networks (Richardson and Domingos, 2006), or MLNs, which learn probabilities attached to first-order formulas. One participant of the second TempEval used a similar approach (Ha et al., 2010). Denis and Muller (2011) cast the problem of learning temporal orderings from texts as a constraint optimization problem. They search for a solution using Integer Linear Programming (ILP), similarly to Bramsen et al. (2006), and Chambers and Jurafsky (2008a). Because ILP is costly (it is NP-hard), the latter two only consider *before* and *after* relations.

Most of these approaches are similar to ours in that they can use knowledge about one TempEval task to solve the other tasks. However, these studies do not report on the full set of logical constraints

```
<TIMEX3 tid="t190" type="TIME" value="1998-02-06T22:19:00"
functionInDocument="CREATION_TIME">06/02/1998 22:19:00</TIMEX3>
<s>WASHINGTON - A economia <EVENT eid="e1">criou</EVENT> empregos a um ritmo surpreendentemente
robusto em <TIMEX3 tid="t191" type="DATE" value="1998-01">janeiro</TIMEX3>, <EVENT
eid="e4">informou</EVENT> o governo na <TIMEX3 tid="t193" type="DATE"
value="1998-02-06">sexta-feira</TIMEX3>, provas de que o vigor económico da América <EVENT
eid="e6">resistiu</EVENT> a todas as <EVENT eid="e7">perturbações</EVENT> <EVENT
eid="e224">causadas</EVENT> até agora pelo <EVENT eid="e228">tumulto</EVENT> financeiro na
Ásia.</s>
<TLINK lid="l1" relType="OVERLAP" eventID="e4" relatedToTime="t193" task="A"/>
<TLINK lid="l2" relType="AFTER" eventID="e4" relatedToTime="t191" task="A"/>
<TLINK lid="l26" relType="BEFORE" eventID="e4" relatedToTime="t190" task="B"/>
```

Figure 2: Example of the Portuguese data used (simplified). The fragment is: *WASHINGTON - A economia criou empregos a um ritmo surpreendentemente robusto em janeiro, informou o governo na sexta-feira, provas de que o vigor económico da América resistiu a todas as perturbações causadas até agora pelo tumulto financeiro na Ásia.*

used or explore little information (e.g. the transitivity of temporal precedence only). Our work does not have these shortcomings: we employ a comprehensive set of reasoning rules (see Section 5.1).

Our approach of encoding in features information that is obtained from automated reasoning does not guarantee that, at the end, the automatically classified temporal relations are consistent. This is a limitation of our approach that is not present in some of the above mentioned work. However, our approach is not sensitive to the size of the training data, since the reasoning rules are hand-coded. With MLNs, even though the rules are also designed by humans, the weight of each rule still has to be learned in training.

One participant of the first TempEval used "world-knowledge axioms" as part of a symbolic solution to this challenge (Puşcaşu, 2007). This world-knowledge component includes rules for reasoning about time. Closest to our work is that of Tatu and Srikanth (2008). The authors employ information about task B and temporal reasoning as a source of classifier features for task C only. This is more limited than our approach: we also explore the other tasks as sources of knowledge, besides task B, and we also experiment with solutions for the other tasks, not just task C.

4 Annotation Scheme and Data

For the experiments reported in this paper we used TimeBankPT (Costa and Branco, 2012), which is an adaptation to Portuguese of the English data used in the first TempEval. These data were produced by translating the English data used in the first TempEval and then adapting the annotations so that they conform to the new language.

Figure 2 shows a sample of that corpus. As before, that figure is simplified. For instance, the full annotation for the first event event term in that example is: `<EVENT eid="e1" class= "OCCURRENCE" stem="criar" aspect="NONE" tense="PPI" polarity="POS" pos="VERB">criou</EVENT>`.

TimeBankPT is similar in size to the English TempEval data. It contains 60K word tokens for training and close to 9K words for evaluation (the word counts are somewhat higher than those for its English counterpart because of language differences). Overall (i.e. for all tasks combined), the number of temporal relations (i.e. instances for classification) is 5,781 for training and 758 for evaluation. The two corpora are quite similar to each other, as one is the translation of the other.

5 Feature Design

The main rationale behind our approach is that, when a system annotates raw text, it may split the annotation process in several steps, corresponding to the different TempEval tasks. In this scenario, the information annotated in previous steps can be used. That is, e.g. if one has already classified the temporal relations between the events in a text and its creation time (task B, which is also the easiest), this information can then be used to help classify the remaining temporal relations.

Our goal is then to evaluate new features for machine learned classifiers for these three tasks. These new features are meant to help predict the class feature by computing the temporal closure of a set of initial temporal relations. This initial set of temporal relations is composed of relations coming from two sources:

- Temporal relations between pairs of dates or times corresponding to annotated temporal expressions. Because the annotations for time expressions contain a normalized representation of them, it is possible to order them symbolically. That is, they are ordered according to the `value` attribute of the corresponding `TIMEX3` element.[3]
- The temporal relations annotated for the other tasks.

The values for these features reflect the possible values of the class feature (i.e. the temporal relation being classified), after applying temporal reasoning to these two sets of relations.

The possible values for these classifier features are the six class values (`BEFORE`, `AFTER`, `OVERLAP`, `BEFORE-OR-OVERLAP`, `OVERLAP-OR-AFTER` and `VAGUE`).[4]

For the sake of experimentation, we try all combinations of tasks:

- Predict task A after temporally closing the relations annotated for tasks B and C (and the temporal relations between the times mentioned in the document). These are the features **Ab** (based on the temporal relations annotated for task B only), **Ac** (based on the relations for task C only) and **Abc** (based on the relations for both tasks).
- Similarly, predict task B, based on tasks A and C: the features **Ba** (based on the relations for task A only), **Bc** (based on the relations for task C only) and **Bac** (based on the relations for both tasks).
- Predict task C after temporally closing the relations annotated for tasks A and B: the features **Ca** (based on the relations for task A only), **Cb** (based on the relations for task B only) and **Cab** (based on the relations for both of them).

The usefulness of these classifier features is limited in that they have very good precision but low recall, as temporal reasoning is unable to restrict the possible type of temporal relation for many instances. In fact, we did not test some of these features, because they produced the `VAGUE` value for all training instances. This was the case of the features **Ac** and **Bc** (and also **Avc** and **Bvc**, which are presented below).

For this reason, we additionally experimented with another set of features that, instead of trying to predict the class value directly, may provide useful heuristics to the classifiers. These are:

- For task B, from all annotated temporal expressions in the same sentence as the event being related to the DCT, the majority temporal relation between those temporal expressions and the DCT, based on their annotated `value` attributes. This is the feature **Bm**.

[3] Chambers and Jurafsky (2008a) also perform this step, but they consider far fewer possible formats of dates and times than we do. The full set of rules used to order times and dates can be found in Costa (2013).

[4] It must be noted that the values `BEFORE-OR-OVERLAP` or `OVERLAP-OR-AFTER` are output when none of the three more specific values (`BEFORE`, `OVERLAP` and `AFTER`) can be identified by the temporal reasoner but one of them can be excluded (i.e. `OVERLAP-OR-AFTER` is used when `BEFORE` can be excluded). Similarly, `VAGUE` is output when no constraint can be identified from the initial set of temporal relations. These underspecified values do not necessarily correspond to the cases when the annotated data contain these values (those are the cases when the human annotators could not agree on a more specific value). It often is the case that the human annotation is more specific, as humans have access to further information.

- For task B, the temporal relation between the time expression closest to the event being ordered with the DCT and the DCT. This is the feature **Bt**.
- A vague temporal relation for task A based on the relations annotated for tasks B and C. These are the classifier features **Avb**, **Avc** and **Avbc**.
- A vague temporal relation for task B based on the relations annotated for tasks A and C: classifier features **Bva**, **Bvc** and **Bvac**.
- A vague temporal relation for task C based on the relations annotated for tasks A and B: features **Cva**, **Cvb** and **Cvab**.

These temporal relations that we call vague are useful when the reasoning component does not identify a precise temporal relation between the two relevant entities in the temporal relation (due to insufficient information). In these cases, it may be useful to known that e.g. both of them temporally overlap a third one, as this may provide some evidence to the classifiers that they are likely to overlap. This is what these vague features encode. Their possible values are: (i) a third entity precedes the two entities, (ii) a third entity overlaps both entities, (iii) a third entity follows the two entities (iv) any combination of any of the above, (v) the first entity in the relation to be guessed overlaps a third entity that temporally follows the second entity in the relation to be guessed, (vi) the first entity in the relation to be guessed overlaps a third entity that temporally precedes the second entity in the relation to be guessed, (vii) the two entities are not even connected in the temporal graph for the document, whose edges correspond to overlap and precedence relations, (viii) none of the above.

5.1 Temporal Reasoning Rules

The rules implemented in our reasoning component are: (i) temporal precedence is transitive, irreflexive and antisymmetric; (ii) temporal overlap is reflexive and symmetric; (iii) if A overlaps B and B precedes C, then C does not precede A.

Because we also consider temporal relations between times and dates, we also deal with temporal inclusion, a type of temporal relation that is not part of the annotations used in the TempEval data, but that is still useful for reasoning. We make use of the following additional rules, dealing with temporal inclusion: (i) temporal inclusion is transitive, reflexive and antisymmetric; (ii) if A includes B, then A and B overlap; (iii) if A includes B and C overlaps B, then C overlaps A; (iv) if A includes B and C precedes A, then C precedes B; (v) if A includes B and A precedes C, then B precedes C; (vi) if A includes B and C precedes B, then either C precedes A or A and C overlap (A cannot precede C); (vii) if A includes B and B precedes C, then either A precedes C or A and C overlap (C cannot precede A).

As mentioned, temporal expressions are ordered according to their normalized value. For instance, the date `2000-01-03` is ordered as preceding the date `2010-03-04`. Since all temporal expressions are normalized in the annotated data, we order temporal expressions before applying any temporal reasoning. This increases the number of temporal relations we start with, and the potential number of relations we end up with after applying temporal reasoning.

To this end, we used Joda-Time 2.0 (`http://joda-time.sourceforge.net`). Each normalized date or time is converted to an interval.

In many cases it is possible to specify the start and end points of this interval, e.g. the date of January 3, 2000 is represented internally by an interval with its start point at `2000-01-03T00:00:00.000` and ending at `2000-01-03T23:59:59.999`. Many different kinds of normalized expressions require many rules. For instance, an expression like *last Winter* could be annotated in the data as `2010-WI`, and dedicated rules are used to get its start and end points.

Some time expressions are normalized as `PRESENT_REF` (e.g. *now*), `PAST_REF` (*the past*) or `FUTURE_REF` (*the future*). These cases are not represented by any Joda-Time object. Instead we need to account for them in a special way. They can be temporally ordered among themselves (e.g. `PRESENT_REF` precedes `FUTURE_REF`), but not with other temporal expressions. We further stipulate

Feature	Task A	Task B	Task C
event-aspect	d--kn	----n	d--kn
event-polarity	d--kn	--r-n	----n
event-POS	--r-n	---k-	----n
event-stem	-jrk-	--r-n	-----
event-string	--r-n	-j---	-----
event-class	djr-n	-jrk-	djrkn
event-tense	--r--	djrkn	djrkn

Feature	Task A	Task B	Task C
o-event-first	djrkn	N/A	N/A
o-event-between	djrkn	N/A	N/A
o-timex3-between	-jrk-	N/A	N/A
o-adjacent	-j--n	N/A	N/A
timex3-mod	----n	---k-	N/A
timex3-type	d-rk-	--rk-	N/A

Table 1: Features used in the baseline classifiers. Key: `d` means the feature is used with `DecisionTable`; `j`, with `J48`; `r`, with `JRip`; `k`, with `KStar`; `n`, with `NaiveBayes`.

that `PRESENT_REF` includes each document's creation time (which therefore precedes `FUTURE_REF`, etc.). So, in additional to the representation of times and dates as time intervals, we employ a layer of *ad-hoc* rules.

The variety of temporal expressions makes it impossible to provide a full account of the implemented rules in this paper, but they are listed in full in Costa (2013).

6 Experiment and Results

Our goal is to test the features introduced in Section 5. Our methodology is to extend existing classifiers for the problem of temporal relation classification with these features, and check whether their performance improves.

For the first TempEval, Hepple et al. (2007) used simple classifiers that use the annotations present in the annotated data as features. They trained Weka (Witten and Frank, 1999) classifiers with these features and obtained competitive results. 10-fold cross-validation on the training data was employed to evaluate different combinations of features.

For our baselines, we use the same approach as Hepple et al. (2007), with the Portuguese data mentioned above in Section 4.

6.1 Experimental Setup

The classifier features used in the baselines are also similar to the ones used by Hepple et al. (2007).

The *event* features correspond to attributes of `EVENT` elements according to the data annotations, with the exception of the `event-string` feature, which takes as value the character data inside the corresponding TimeML `EVENT` element. In a similar fashion, the *timex3* features are taken from the attributes of `TIMEX3` elements with the same name.

The *o(rder)* features are the attributes computed from the document's textual content. The feature `order-event-first` encodes whether the event terms precedes in the text the time expression it is related to by the temporal relation to classify. The classifier feature `order-event-between` describes whether any other event is mentioned in the text between the two expressions for the entities that are in the temporal relation, and similarly `order-timex3-between` is about whether there is an intervening temporal expression. Finally, `order-adjacent` is true if and only if both `order-timex3-between` and `order-event-between` are false (even if other linguistic material occurs between the expressions denoting the two entities in the temporal relation).

Just like Hepple et al. (2007), we experimented with several machine learning algorithms. Table 1 shows the classifier features that we selected for each algorithm. For each algorithm and task, we tried all possible combinations of features and selected the one that performed best, according to 10-fold cross-validation on the training data.

	Task A		Task B		Task C	
Classifier	bl.	best	bl.	best	bl.	best
DecTable	52.1	**58.6** (Ab,Abc)	77.0	77.0	49.6	49.6 (Cva)
J48	55.6	**58.0** (Ab,Avb)	77.3	**77.9** (Ba,Bva)	52.7	52.7
JRip	59.2	**68.0** (Ab,Avbc)	72.8	**76.7** (Bt,Ba,Bva)	54.3	54.3 (Ca,Cva,Cb,Cab)
KStar	54.4	**59.8** (Ab,Avb,Abc)	73.4	72.8 (Ba,Bva)	53.1	**53.9** (Cva,Cb)
NBayes	53.3	**56.2** (Ab,Avb)	75.2	**75.3** (Ba)	53.9	53.5 (Ca,Cva)
Average	54.9	**60.1**	75.1	**75.9**	52.7	**52.8**

Table 2: Classifier accuracy on test data (bl.: baseline; best: baseline extended with best combination of the new features, shown in parentheses, determined with cross-validation on train data). Boldface highlights improvements on test data.

We essentially used the same algorithms as Hepple et al. (2007).[5] We also experimented with J48 (Weka's implementation of the C4.5 algorithm). The classifiers obtained this way are used as baselines. To compare them with solutions incorporating temporal reasoning, we retrained them with the entire training data and evaluated them on the held-out test data. The results are shown in the columns of Table 2 labeled with *bl.* (baselines). We chose these baselines because they are very easy to reproduce: the algorithms are open-source and the classifier features are straightforwardly extractable from the annotated data and only require simple string manipulation.

For each task (A, B and C) and algorithm, we extended the best classifier previously found with the features that were presented above in Section 5. We kept the basic features, listed in Table 1 (i.e. the ones selected in the manner just reported), constant and tried all combinations of the new features, based on temporal reasoning. We then selected the feature combination that produced the best results for each algorithm and task, using 10-fold cross-validation on the train data, and, once again, evaluated the combination thus chosen on the held-out test data.

6.2 Results and Discussion

The results can be seen in Table 2. They vary by task. The tested classifier features are quite effective for task A. The new features are, however, much less effective for the other tasks. This is perhaps more surprising in the case of task C. It is mostly a problem with recall (the new reasoning-based features are able to restrict the possible type of temporal relation only for a few instances, because the data are not very densely annotated for temporal relations). That is, reasoning is very precise but leaves many instances unaccounted for. For instance, out of 1735 train instances for task C, 1589 have the value VAGUE for the feature **Cb**. In the test data, this is 241 instances out of 258.

For task A, we inspected the final decision tree (obtained with J48), the decision table (DecisionTable) and the rules (JRip) induced by the learning algorithms from the entire training set. The tree for task A checks the feature **Ab** and outputs the same type of temporal relation as the one encoded in that feature. When the value of this feature is one of the disjunctive values (VAGUE, BEFORE-OR-OVERLAP and OVERLAP-OR-AFTER), it consults the remaining features. Because of the way that trees are built by this algorithm (J48, an implementation of the C4.5 algorithm), this means that the feature **Ab** is the classifier feature with the highest information gain, among those used by this classifier. The same feature **Ab** appears frequently in the antecedent of the rules induced by JRip for task A (it occurs in the antecedent of 5 of the 8 induced rules), another indication that it is quite useful. When learning a table that associates combinations of feature values with class values, the DecisionTable algorithm

[5]These are: Weka's implementation of decision tables, Dec(ision)Table; the RIPPER algorithm, JRip; N(aive)Bayes, a Bayesian classifier; and KStar, a k-NN algorithm with an entropy-based distance function. We left out support vector machines, which are too slow for exhaustive search to be practical, even with this limited set of features. Hepple et al. (2007) tried this algorithm, but selected classifier features using a greedy search method.

	Task A		Task B		Task C	
Classifier	bl.	best	bl.	best	bl.	best
DecTable	52.1	**54.4** (Ab,Abc,Avbc)	77.0	77.0	49.6	49.6 (Cvb,Cvab)
J48	55.6	54.4 (Ab,Avb)	77.3	**79.5** (Ba)	52.7	51.9 (Cvb)
JRip	59.2	**64.5** (Avb,Abc)	72.8	**74.0** (Bt,Ba,Bac,Bvac)	54.3	54.3
KStar	54.4	**58.6** (Ab,Avb)	73.4	71.9 (Bva)	53.1	52.7 (Cva)
NBayes	53.3	**55.6** (Ab,Avbc)	75.2	**75.5** (Bm,Bac)	53.9	**54.3** (Cb)
Average	54.9	**57.5**	75.1	**75.6**	52.7	52.6

Table 3: Classifier accuracy on test data, with the reasoning-based features computed from the temporal relations classified by the baseline classifiers.

prunes some of the classifier features: the feature **Abc** is pruned, but the feature **Ab** is kept, another indication that task B relations are useful when classifying task A relations.

Inspection of the learned models thus suggests that information about task C is not as useful to solve task A as the information coming from task B. This is easy to understand: task A relates entities in the same sentence, whereas task C relates entities in different sentences; they also relate different kinds of entities (task C temporal relations are between two events whereas task A relations are between an event and a time). As such, temporal relations with arguments in common are not found between these two tasks, and only long chains of relations can support inferences,[6] but they are infrequent in the data.

The results in Table 2 are obtained with reasoning based on the gold standard annotations. That is, a feature such as **Ab** tries to predict the class of task A relations on the basis of task B temporal relations, and these task B relations are taken from the gold standard. In a real system, we do not have access to this information. Instead, we have temporal relations classified with some amount of error. We would have to look at the output of a classifier for task B in order to compute this feature **Ab**. An interesting question is thus how our approach performs when the initial temporal relations given to the reasoning component are automatically obtained. Table 3 presents these results. In this table, the reasoning component acts on the output of the baseline classifiers. For instance, the feature **Ab** tries to predict task A temporal relations using the reasoning rules on the output of the corresponding baseline classifier for task B (i.e. task B temporal relations that have been automatically classified by the baseline classifier employing the same learning algorithm).[7]

As can be seen from Table 2, the results are slightly worse, but there is still a noticeable and systematic improvement in task A. Under both conditions (Table 2 and Table 3), the differences between the baseline classifiers and the classifiers with the new features are statistically significant for task A ($p < 0.05$, according to Weka's `PairedCorrectedTTester`), but not for the other tasks. For this task at least, reasoning is a useful means to improve the temporal relation classification. Comparing the two tables, we can conclude that as temporal relation classification improves (and the error present in the initial temporal relations on which reasoning is based goes down), so does the positive impact of reasoning increase: the results in Table 2 are better than the ones in Table 3 because the initial temporal relations on which temporal reasoning is based are better quality. Therefore, as the performance of existing temporal relation classification technology improves, so should the potential impact of these features based on reasoning. Another conclusion is that, even with the current technology, these features are already useful, as Table 3 presents statistically significant improvements on task A.

In a real system for temporal processing, these new features cannot be used for all tasks. When temporally annotating text automatically, assuming one classifier for each task, one must choose an order

[6]For instance, according to task C, an event e_1 precedes another event e_2, which precedes the document creation time according to task B, which precedes a time t_3 according to their annotated `value`, therefore event e_1 must precede t_3.

[7]In this case, the input relations may be inconsistent. We can detect sets of inconsistent temporal relations, but we cannot know which temporal relations in such a set are misclassified. For this reason, we simply add temporal relations to the reasoning component according to textual order, and a relation is skipped if it is inconsistent with the previously added ones.

of processing the three tasks, and this determines which features are available for each classifier. Since task A benefits considerably from these features, a practical system incorporating our proposal would classify the temporal relations for tasks B and C first (taking advantage of none of the new features, as they do not improve these two tasks), and then a classifier for task A, trained using these new features, can be run, based on the output for the other tasks.

7 Concluding Remarks

In this paper we showed that features based on logical information improve existing classifiers for the problem of temporal information processing in general and temporal relation classification in particular. Even though temporal reasoning has been used in the context of temporal information processing to oversample the data (Mani et al., 2006), to check inter-annotator agreement (Setzer and Gaizauskas, 2001), as part of an annotation platform (Verhagen, 2005), or as part of symbolic approaches to the TempEval problems (Puşcaşu, 2007), to the best of our knowledge the present paper is the first to report on the use temporal reasoning as a systematic source of features for machine learned classifiers.

References

Ahn, D., S. Schockaert, M. D. Cock, and E. Kerre (2006). Supporting temporal question answering: Strategies for offline data collection. In *5th International Workshop on Inference in Computational Semantics*, Buxton.

Allen, J. (1984). Towards a general theory of action and time. *Artificiall Intelligence 23*, 123–154.

Bejan, C. A. and S. Harabagiu (2010). Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the ACL*, Uppsala, pp. 1412–1422. ACL.

Bramsen, P., P. Deshpande, Y. K. Lee, and R. Barzilay (2006). Inducing temporal graphs. In *Proceedings of EMNLP 2006*, Sydney, pp. 189–198.

Chambers, N. and D. Jurafsky (2008a). Jointly combining implicit constraints improves temporal ordering. In *Proceedings of EMNLP 2008*, Honolulu, pp. 698–706. ACL.

Chambers, N. and D. Jurafsky (2008b). Unsupervised learning of narrative event chains. In *Procedings of the 46th Annual Meeting of the ACL*, Columbus, pp. 789–797. ACL.

Costa, F. (2013). *Processing Temporal Information in Unstructured Documents*. Ph. D. thesis, Universidade de Lisboa, Lisbon. To appear.

Costa, F. and A. Branco (2012). TimeBankPT: A TimeML annotated corpus of Portuguese. In *Proceedings of LREC 2012*, Istanbul, pp. 3727–3734. ELRA.

Denis, P. and P. Muller (2011). Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In *Proceedings of IJCAI 2011*.

Freksa, C. (1992). Temporal reasoning based on semi-intervals. *Artificial Intelligence 54*(1), 199–227.

Ha, E. Y., A. Baikadi, C. Licata, and J. C. Lester (2010). NCSU: Modeling temporal relations with Markov logic and lexical ontology. In *Proceedings of SemEval 2010*, Uppsala, pp. 341–344. ACL.

Hepple, M., A. Setzer, and R. Gaizauskas (2007). USFD: Preliminary exploration of features and classifiers for the TempEval-2007 tasks. In *Proceedings of SemEval-2007*, Prague, pp. 484–487. ACL.

Katz, G. and F. Arosio (2001). The annotation of temporal information in natural language sentences. In *Proceedings of the 2001 ACL Workshop on Temporal and Spatial Information Processing*, Toulouse.

Ling, X. and D. S. Weld (2010). Temporal information extraction. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*.

Mani, I., M. Verhagen, B. Wellner, C. M. Lee, and J. Pustejovsky (2006). Machine learning of temporal relations. In *Proceedings of the 44th Annual Meeting of the ACL*, Sydney. ACL.

Mani, I., B. Wellner, M. Verhagen, and J. Pustejovsky (2007). Three approaches to learning TLINKs in TimeML. Technical Report CS-07-268, Brandeis University.

Pan, F., R. Mulkar-Mehta, and J. R. Hobbs (2011). Annotating and learning event durations in text. *Computational Linguistics 37*(4), 727–752.

Puşcaşu, G. (2007). WVALI: Temporal relation identification by syntactico-semantic analysis. In *Proceedings of SemEval-2007*, Prague, pp. 484–487. ACL.

Pustejovsky, J., J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, and G. Katz (2003). TimeML: Robust specification of event and temporal expressions in text. In *IWCS-5, Fifth International Workshop on Computational Semantics*.

Pustejovsky, J., P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo (2003). The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*.

Regneri, M., A. Koller, and M. Pinkal (2010). Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the ACL*, Uppsala, pp. 979–988. ACL.

Richardson, M. and P. Domingos (2006). Markov logic networks. *Machine Learning 62*(1), 107–136.

Saquete, E., P. Martínez-Barco, R. Muñoz, and J. L. Vicedo (2004). Splitting complex temporal questions for question answering systems. In *Proceedings of the 42nd Meeting of the ACL*, Barcelona. ACL.

Setzer, A. and R. Gaizauskas (2001). A pilot study on annotating temporal relations in text. In *ACL 2001 Workshop on Temporal and Spatial Information Processing*.

Tao, C., H. R. Solbrig, D. K. Sharma, W.-Q. Wei, G. K. Savova, and C. G. Chute (2010). Time-oriented question answering from clinical narratives using semantic-web techniques. In *Proceedings of the 9th International Conference on the Semantic Web*, Volume 2, Berlin, pp. 241–256.

Tatu, M. and M. Srikanth (2008). Experiments with reasoning for temporal relations between events. In *Proceedings of COLING 2008*, Volume 1.

Vendler, Z. (1967). Verbs and times. In *Linguistics in Philosophy*, pp. 97–121. Ithaca, New York: Cornell University Press.

Verhagen, M. (2005). Temporal closure in an annotation environment. In *Language Resources and Evaluation*, Number 39, pp. 211–241.

Verhagen, M., R. Saurí, T. Caselli, and J. Pustejovsky (2010). SemEval-2010 task 13: TempEval-2. In *Proceedings of SemEval-2010*.

Vilain, M., H. Kautz, and P. van Beek (1990). Constraint propagation algorithms for temporal reasoning: A revised report. In *Readings in Qualitative Reasoning about Physical Systems*, pp. 373–381. San Francisco: Morgan Kaufmann.

Witten, I. H. and E. Frank (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco: Morgan Kaufmann.

Yoshikawa, K., S. Riedel, M. Asahara, and Y. Matsumoto (2009). Jointly identifying temporal relations with Markov logic. In *Proceedings of the 47th Annual Meeting of the ACL*.