Experiments with Reasoning for Temporal Relations between Events

Marta Tatu and **Munirathnam Srikanth**
Lymba Corporation
Richardson, Texas, United States
marta,srikanth@lymba.com

Abstract

Few attempts have been made to investigate the utility of temporal reasoning within machine learning frameworks for temporal relation classification between events in news articles. This paper presents three settings where temporal reasoning aids machine learned classifiers of temporal relations: (1) expansion of the dataset used for learning; (2) detection of inconsistencies among the automatically identified relations; and (3) selection among multiple temporal relations. Feature engineering is another effort in our work to improve classification accuracy.

1 Introduction

In recent years, there has been a growing interest in temporal information extraction, as more and more operational natural language processing (NLP) systems demand dealing with time-related issues in natural language texts. Machine learning-based temporal relation identification has been explored by only a few researchers, including Boguraev and Ando (2005), Mani et al. (2006), Chambers et al. (2007), and the TempEval 2007 participants (Verhagen et al., 2007).

For a given ordered pair of elements (x_1, x_2), where x_1 and x_2 are events or times, temporal relation resolution is the task of automatic identification of the relation $r_i \in TempRel$ that temporally links x_1 and x_2. For example, given the statement *Mr. Antar was charged*$_{e_{137}}$ *last month*$_{t_{237}}$ *in a civil suit*$_{e_{138}}$ *filed*$_{e_{140}}$ *in federal court in Newark by the Securities and Exchange Commission* (wsj_0778[1]) and the pairs (e_{137}, t_{137}), (e_{137}, e_{138}), and (e_{138}, e_{140}), the task is to automatically label the given pairs with the *is_included*, *is_included*, and *simultaneous* relations, respectively. We note that the granularity of the temporal relations ($TempRel$) varies from TimeML's 14 relations to TempEval's three coarse-grain relations.

While machine learning approaches attempt to improve classification accuracy through feature engineering, Mani et al. (2006) introduced a temporal reasoning component to greatly expand the training data. By computing the temporal closure of the training data relations, they increased the training set by a factor of 10. They reported encouraging accuracy of classification on event-event and event-time relations. According to their experiments, the event-event relation accuracy goes from 62.5% to 94.95% and the event-time relation accuracy ranges from 73.68% to 90.16%. Recently, extensions of Mani et al. (2006)'s research is briefly described in (Mani et al., 2007). This technical report addresses two problems found in (Mani et al., 2006): (1) feature vector duplication caused by the data normalization process (once fixed, the accuracy drops to 76.56% and 83.23%) and (2) a somewhat unrealistic evaluation scheme (we describe Mani et al. (2007)'s results in Section 4.1).

TempEval 2007 is the first standard evaluation arena that consists of three temporal relation classification tasks (Verhagen et al., 2007). The participants reported F-measure scores ranging from 42% to 55% for event-event relations, and 73% to 80% for event-time relations.

Because of their different experimental settings,

[1] All examples shown here are taken from TimeBank 1.2.

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 857–864
Manchester, August 2008

the results reported in (Mani et al., 2007) cannot be directly compared with those of TempEval 2007 participants. Among others, the three major differences are:

1. significantly different training and testing data. Although the datasets used TimeBank 1.2 (Pustejovsky et al., 2003), Mani et al. (2007) added the AQUAINT Corpus (www.timeml.org) to their experimental data;
2. different sets of temporal relations. Mani et al. (2006; 2007) target six normalized relations (*before*, *immediately before* (*ibefore*), *includes*, *ends*, *begins* and *simultaneous*). In TempEval 2007, a set of three coarse-grain temporal relations was used (*before*, *after*, and *overlap*).
3. different relation scope. In (Mani et al., 2006; Mani et al., 2007), event-event temporal relations are discourse-wide, i.e. *any* pair of events can be temporally linked. For TempEval 2007, the event-event relations are restricted to events within two consecutive sentences.

These two modeling frameworks for solving the problem of temporal relation classification produce highly dissimilar results. With this in mind, we are interested in two issues in this paper: (1) *How might temporal reasoning assist in temporal relation identification?* (2) *What other features might be used to improve the performance of classification?* As a byproduct of our exploration to these two questions, we hope to find some insights on why the same problem explored under different environment produces highly divergent results.

In this paper, we investigate several interactions between temporal reasoning and a machine learning approach for temporal ordering of events in natural language texts. We continue by describing the data used for our experiments. In Section 3, we briefly describe the set of features we currently use to build Support Vector Machine (SVM) (Chang and Lin, 2001) and Maximum Entropy (ME) models for temporal relation resolution. The three interactions we envision between temporal reasoning and the learned models are presented in Section 4. In conclusion, we present a discussion of our experimental results and future research directions.

2 Data Preparation and Analysis

2.1 TimeBank 1.2

In this paper, we use the TimeBank 1.2 data (Pustejovsky et al., 2003). This is the first attempt to create a corpus with human annotated temporal relations. It contains 183 news documents collected from several news agencies.

2.2 Data normalization

Similar to (Mani et al., 2006; Mani et al., 2007), we use a normalized version of the 14 temporal relations annotated in TimeBank where the inverse relations are removed and *simultaneous* and *identity* are collapsed as well as *during* and *is_included*. The distribution of the normalized event-event temporal relations annotated in the data we used for training our temporal resolution models is shown in Table 2.

2.3 Experimental data

For the experiments described in this paper, we used a random 80-20 percent split of the TimeBank data to train and test the learned classifiers (36 randomly selected documents for testing and the remaining 147 for training the models) and 5-fold-cross-validation of the training data for parameter tuning. We note that our experimental setup is closer to the one used in (Mani et al., 2006; Mani et al., 2007). Noting that we do not use the AQUAINT Corpus in our experiments, our results can be compared with theirs, but not with the TempEval system performances.

3 Feature Engineering

As reported by participants in TempEval 2007 (Verhagen et al., 2007), (Boguraev and Ando, 2005), (Chambers et al., 2007), and (Mani et al., 2007), most of the features used for learning are syntactic attributes extracted for single event terms. In our work, we have experimented with *semantic* features and attributes which take the *event's linguistic context* into consideration (Min et al., 2007). Our experiments show that only few features are critical and impact the classifier's accuracy (Table 1). These include the basic features available in TimeBank, e.g. *event-class*, *tense*, *aspect*, *polarity*, *modality*, *stem*, and *part of speech* of event terms (*Baseline* row in Table 1). Additional features that we explored include:

Feature set	Accuracy (%)
Baseline	46.3
Baseline with sameActor	+0.4
Baseline with eventCoref	+0.2
Baseline with oneSent	+4.0
Baseline with relToDocDate	+0.2
Baseline with tensebigram	+0.8
Baseline with tensetrigram	+0.6
Baseline with all	57.4

Table 1: New features impact

1. *sameActor*. This binary feature indicates whether the two events share the same semantic role AGENT. The motivation behind this feature is that two event terms, especially, verbs, which have the same agent, have a closer semantic relationship and, accordingly, are temporally related.

2. *eventCoref*. This binary attribute captures the event co-reference information. If two events co-refer, even though they have different surface forms, they must take place simultaneously. For instance, the *offer* and *deal* events, mentioned in the following sentences, refer to the same *transaction* and, therefore, must be linked by a *simultaneous* relation.

   *a) Sony Corp. completed its tender **offer** for Columbia Pictures Entertainment Inc., with Columbia shareholders tendering 99.3% of all common shares outstanding by the Tuesday deadline.*

   *b) Sony Columbia Acquisition Corp., formed for the Columbia **deal**, will formally take ownership of the movie studio later this month, a spokesman said.*

3. *oneSent*. This binary feature is true if the two events are part of the same sentence. Chambers et al.(2007) introduced this feature in their experiments, and our analysis shows that this attribute has a relatively larger contribution to the overall performance. The intuition behind this feature is that the closer two events get, the closer their temporal relationship is.

4. *relToDocDate*. This feature encodes the temporal relation between each event and the Document Date. This was one of the subtasks of TempEval 2007 and we used this relationship as a feature. Our motivation is that we might be able to infer the relationship between two events e_1 and e_2 from the temporal relations they have with the Document Date. For example, if $before(e_1, DocDate)$ and $after(e_2, DocDate)$ are true, then $before(e_1, e_2)$. There may be two reasons for the low impact of this feature: (1) an accurate computation of the temporal relation between an event and the Document Date is not easy, as demonstrated in TempEval 2007 and (2) if two events have the same relation with the Document Date, there is no way to determine the event-event relation.

5. *tenseBigram* and *tenseTrigram*. Going beyond using the *tense* attribute for single event terms, we extract bigrams and trigrams with the tense values of the current event and immediately preceding and following events. This feature is intended to reflect the tense shifts of sequential events as part of a larger context of the current event.

All these features have a positive impact on the performance of the learned classifiers (Table 1). Further improvement is desired and we use temporal reasoning in three different settings in an attempt to obtain more accurate temporal relations.

4 Temporal Reasoning

Following our feature set improvements for machine learned models of temporal relations, we turned to *temporal reasoning* and explored different ways in which it can aid the resolution of temporal relations. We experimented with three different interactions between our temporal reasoning and temporal relation resolution modules.

Our natural language reasoning engine (Tatu and Moldovan, 2005; Tatu and Moldovan, 2007) makes use of (1) a first-order logical representation of the input document which captures the concepts mentioned in the text, their attributes including named entity class values, event class or normalized values (for times) and the syntactic as well as the semantic dependencies between concepts[2]; (2) a rich set of axioms which encode the knowledge needed to derive meaningful information from a document; (3) a logic prover which operates in a proof by contradiction manner (a hypothesis H is entailed by a text T assumed to be true, denoted by $T \vdash H$, if and only if $(T \wedge \neg H) \vdash \bot$, where $\bot$ is *false*). Given the logical transformation of a text T, the prover uses the knowledge encoded in the

[2] These dependencies include the temporal relations identified either by human annotators or by the models presented in Section 3.

axioms (Bk) to derive new information ($T^\star$) about T^3 and scores the best mapping of the hypothesis H to $T^\star$.

For the temporal relation resolution experiments presented in this paper, we are interested in deriving additional temporal information from an input document without checking the entailment between this document and a hypothesis. Therefore, for the following tasks, the text T is a TimeBank 1.2 document and the set of axioms Bk used by the prover contains 94 temporal axioms which link each temporal relation with its inverse ($R^{-1}(x,y) \leftrightarrow R(y,x)$, for example $before(x,y) \rightarrow after(y,x)$) and define the temporal relation resulting from the combination of two relations ($R_1(x,y) \wedge R_2(y,z) \rightarrow R_3(x,z)$, for example, $before(x,y) \wedge before(y,z) \rightarrow before(x,z)$). These axioms were derived from Allen's interval algebra (Allen, 1983).

We note that the prover computes and uses temporal relations between any two temporal expressions mentioned in an input TimeBank document[4] (e.g. *now* [19891101] and *last year* [1988] are linked by a *before* temporal relation in *wsj_0324*).

Within this reasoning setup, the information derived by the prover ($T^\star$) will include the **temporal closure** of the input text's relations. We note that the temporal closure includes event-event, event-time and time-time temporal relations. We also note that the temporal axioms are considered 100% accurate (if the temporal relations given as input are correct, then the temporal relations derived using the axioms are also correct).

4.1 Training data expansion

Our first effort to create more accurate temporal relation resolution classifiers given our temporal reasoning engine is to *augment* the gold training data with new relations from the temporal closure of the relations identified by human annotators. Therefore, given the 3,527 temporal relations annotated in the TimeBank data used to train our initial temporal resolution models, we derived 12,270 new relations (an increase of 3.47 times). We show in Table 2 statistics of the normalized event-event relations for both the original and the closed training data. We note that the temporal inconsistencies identified in the original training data (by the procedure described in Section 4.2) were resolved manually by one of the authors of this paper.

Relation	Original data		Closed (*) data	
	Freq.	%	Freq.	%
ibefore	51	2.06	137	1.59
begins	52	2.10	119	1.38
ends	61	2.47	125	1.45
includes	434	17.59	1,161	13.47
before	885	35.88	3,165	36.73
simultaneous	983	39.86	3,909	45.36
Total	2,466	100.00	8,616	100.00

Table 2: Normalized training data (event-event relations)

We built SVM and ME models from the total of 8,616 normalized temporal relations using the set of 15 features described in Section 3. Table 3 shows the performance of the learned models on the test data (original data as well as closed test data). Unlike (Mani et al., 2006), the accuracy of the learned classifiers drops when they are trained on the closed training dataset. By analyzing the results from Table 3, one cannot help but notice the high accuracy on the data used for training and the significant difference between the performance on the training and testing datasets. This may suggest that (1) the machine learners *overfit* the models on the training data and they are not able to generalize and resolve the relations in the test data[5] or (2) the two datasets are very different (in terms of feature values) and the data split happened to create a training data which is not (fully) characteristic to the problem we are trying to solve (the two datasets have different distributions). Therefore, we measured the accuracy of ME models for event-event relation resolution using 5-fold-cross-validation of the entire TimeBank data (Table 4). For these experiments, each TimeBank document (with all its temporal relations) was used as part

Training data	Accuracy for event-event relations		
	Original test (845)	Closed test (4,189)	Train
	ME models		
Original (2,466)	50.4	46.1	83.3
Closed (8,616)	47.0	41.0	76.1
	SVM models		
Original (2,466)	56.9	45.8	74.2
Closed (8,616)	52.4	52.0	77.5

Table 3: Event-event temporal resolution

[3] $T^\star$ contains **all** the information the prover can derive from T given the axioms Bk.

[4] We restricted the time-time relations to only *before*, *simultaneous*, and *after*.

[5] The accuracy of the SVM models is lower on the training data when compared with the ME models while their performance on the test dataset is better.

	Test data 1/5 of the data	**Train data** remaining 4/5s
5-fold-cross split at the *document* level		
Original (3,311)	57.4	89.5
Closed (11,530)	58.2	85.5
5-fold-cross split at the *relation* level		
Original (3,311)	58.3	90.0
Closed (11,530)	73.4	85.3

Table 4: Average ME accuracy for event-event relations using 5-fold-cross-validation on the entire TimeBank data

of either the training or the testing dataset. Our results for the random 5-fold-cross split of the data *at the document level* are similar to the ones obtained for the models learned on the pre-established training data (top two rows in Table 4). Thus, our initial split of the data was not an 'unfortunate' division. The same significant difference between the performance on the unseen data and the training set can be seen. This suggests that some overfitting occurs. Features, such as the event term *stem*, with a large number of possible values mislead the machine learning algorithms and the models they create are not able to correctly classify event pairs with unseen values for these high-valued features. For instance, *showdown* is part of a single TimeBank document (*AP900816-0139*) and the models learned using other documents will misclassify *showdown*'s temporal relations. We note that, by expanding the training data using its temporal closure, *no* new events are added to the training set, only new temporal relations between the *same* set of events are added. Long-term solutions include (1) the expansion of the annotated data or (2) the reduction in the number of values for certain features (for example, by generalizing the event term stem to its WordNet hypernym). In an attempt to homogenize the feature values for the training and the testing datasets, we split the set of normalized event-event temporal relations annotated in TimeBank into training and testing *without* considering the document boundaries. The performance of the learned classifiers increases by 1% when trained on unclosed data and by more than 15% when the closed data is used (Table 4).

In their most recent technical paper, Mani et al. (2007) revise their evaluation method and report performance values for classifiers learned by partitioning the data at the document level (accuracy drops from 59.68% to 51.14% when closed training data is used). These results are consistent with our findings. In the near future, we shall experiment with the second solution we propose above.

4.2 Testing data validation

Given that almost half of the temporal relations automatically identified for the testing data are incorrect when compared to the gold annotation, we decided, as our second experiment, to use temporal reasoning to find *temporal inconsistencies* and replace some of the relations contributing to the inconsistency by the immediate lower confidence relation returned by the learned classifiers. For this purpose, we use an additional set of 77 temporal axioms which encode the *irreflexivity* of temporal relations ($\neg R(x, x)$, for example, $\neg before(x, x)$) and their *empty intersections* ($R_1(x, y) \rightarrow \neg R_2(x, y)$ when $R_1 \neq R_2$, for example $before(x, y) \rightarrow \neg simultaneous(x, y)$).

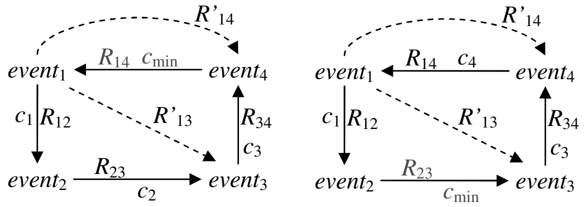
Our process of testing data validation is iterative. Once a temporal inconsistency is identified in the test data, it is resolved and the procedure which computes the temporal closure is re-started. A temporal inconsistency in a TimeBank document (T) is detected every time $\perp \in T^\star$. The automatically identified temporal relations (part of the text T) which contributed to the derivation of $\perp$ become candidates for the resolution of the inconsistency[6]. These candidates are sorted based on the confidence assigned by the machine learning algorithm[7] and the lowest confidence relation is replaced either by the temporal relation found by the prover which directly contradicted the automatically identified relation[8] (Figure 1(a)) or, for the cases where such a relation does not exist, by the immediate lower confidence relation identified by the learned models (Figure 1(b)).

If, for example, for the statement *The US is bolstering its military presence in the gulf, as President Clinton discussed*$_{e_1}$ *the Iraq crisis with the one ally who has backed*$_{e_2}$ *his threat*$_{e_3}$ *of force, British prime minister Tony Blair*, the ME classifier built in Section 3 identifies the temporal relations $before(e_2, e_1)$ (confidence: 0.53), $before(e_3, e_2)$ (0.47) and $includes(e_3, e_1)$ (0.42), the prover identifies the temporal inconsistency

[6] The temporal closure axioms are accurate and do not 'introduce' incorrect temporal relations.

[7] For all experiments which exploit the confidence assigned by the machine learning algorithm, we use the learned ME models (SVM models do not provide a confidence for their decision).

[8] The confidence of a relation derived by the prover is the average of the its 'parent's confidence values.



(a) R_{14} replaced by R'_{14}

(b) R_{23} replaced by the next best relation identified for $(event_2, event_3)$

Figure 1: Temporal inconsistency resolution

generated by these three relations and replaces the lowest confidence relation ($includes(e_3, e_1)$) with the relation it derives from the closure of the other two relations ($before(e_3, e_1)$, confidence: 0.50).

We note that, during the inconsistency checking process, all types of temporal relations are used (event-event, event-time and time-time). For this inconsistency resolution process, we make the assumption that only *one* of the temporal relations which generated the inconsistency is incorrect and should be replaced.

For the testing dataset described in Section 2.3, the validation algorithm found inconsistencies in only 25% of the test documents. This is not very encouraging, given that the accuracy of the temporal relations identified in the other 75% of the documents is 50.4%. The documents marked as inconsistent include, on average, 3.66 temporal inconsistencies (with a maximum of 8 in a single document). For each pair of events, we considered only the top three temporal relations (in terms of confidence) identified by the learned classifiers. When the third relation identified for a given pair of events had to be removed by the inconsistency resolution algorithm, no other temporal relation was added to replace it. Table 5 shows the impact of the validation step on the unclosed test data.

	Precision	Recall	F-measure
Baseline	50.4	50.4	50.4
With test validation	50.1	49.7	49.9

Table 5: Performance change after the testing data validation step. The baseline is the ME model learned on the original (unclosed) training data.

Our error analysis shows that, for each discovered temporal inconsistency, more than one incorrect relation lead to an inconsistent temporal closure. Frequently enough, replacing the lowest confidence relation does not resolve the inconsistency and the temporal relations used to replace it are, in turn, replaced in the next iterations (the confidence of the replacing relation is lower than the confidence of the replaced relation). The ME classifier's numerous errors and the low applicability of this process make its contribution to the overall temporal relation resolution process negative. Our future work will focus on (1) experimenting with less erroneous data for which our one-incorrect-relation-per-inconsistency assumption holds ((perhaps) the models learned from closed training data using a data split at the relation level) and (2) testing the existence of a consistent temporal closure in the absence of the lowest confidence relation. If none of the six temporal relations that we use to label an event-event relation can replace the lowest confidence relation and lead to a consistent temporal closure, then our candidate incorrect relation is among the other higher confidence relations. We also note that we rely heavily on the confidences automatically assigned by the ME classifiers.

Mani et al. (2007) briefly describe a Greedy method for ensuring global consistency of automatically labeled testing data. No evaluation results are reported. As far as we can tell, Mani at el. (2007) use this algorithm to decide whether or not to assign the top 1 relation automatically identified by ME classifiers to a given pair of events. No attempts are made to replace this relation. Our validation algorithm uses lower confidence relations found by the learned models for the same pair of events to replace the lowest confidence relation that leads to a temporal inconsistency.

4.3 Temporal relation alignment

In the previous section, we used temporal reasoning to replace certain relations automatically identified by the learned temporal relation resolution models with the next best relation (in terms of the confidence) found for the same pair of events. For our third experiment, we use the *top* n relations automatically identified for a single pair of events. Across a document, these relationships can be grouped to form different temporal orderings of the events mentioned in the document. For instance, for four pairs of events, 81 different temporal settings can be created using the top 3 temporal relations. Figure 2 shows two of these 81 facets ($\{R_{12}, R'_{23}, R'_{34}, R_{41}\}$ and $\{R"_{12}, R_{23}, R"_{34}, R"_{41}\}$) for events $e_1, \ldots, e_4$.

For these event temporal orderings, we pro-

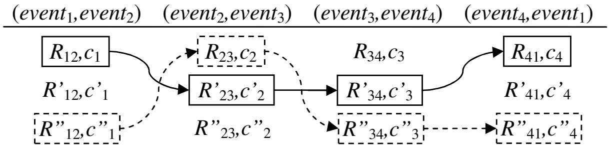


Figure 2: Two possible relation alignments

pose to use our temporal reasoning module to derive, score, and rank their temporal closures. We make the assumption that the correct document-level event ordering, the document's temporal cohesion can be identified by measuring the closure of the document's temporal relations and that orderings that use incorrect relations do not generate good closures. Thus, the relations that generate the best temporal closure will be considered final and will be used to label the test document's event-event and event-time pairs.

For the example shown in Figure 2, 81 different temporal closures are generated depending on the set of relations used to derive them from. In order to find the final four temporal relations between events $e_1, \ldots, e_4$, we score and order the derived temporal closures. The best closure decides, for each pair of events, which relation among the top 3 should be selected as final.

Our first step is to identify the best value for n. Table 6 shows the *maximum* gain in performance when multiple relations are considered for the same pair of events (an instance is considered correct if the gold annotation is among the top n relations returned by the system).

Top n relations	**Accuracy (%)**
1	57.40
2	80.66
3	94.49
4	97.38
5	99.12
6	100.00

Table 6: Top n oracle performance using 5-fold-cross-validation on the TimeBank data

Because there is substantial improvement in the top 3 relation set, we use for our experiments the first three relations identified by the ME classifiers. But, if we consider the top 3 relations for each pair of events in a document, we end up with 3^N possible alignments, where N is the number of event-event and event-time pairs[9] and the scoring and ranking of all 3^N temporal closures becomes hardly possible. Therefore, we use a more *Greedy* approach. Iteratively, we score and rank temporal closures derived from a small set of top 3 relations between N' event-event pairs ($N' < N$) and any final temporal relations. The best closure is used to decide on N' temporal relations which will be added to the best partial alignment and will be used to compute all the following temporal closures.

	Accuracy
Baseline - top 1	50.4
Oracle (upper bound) - top 3	92.4
With test alignment	47.5

Table 7: Test dataset performance change after the testing data alignment step. The top 1 and top 3 baselines were generated using the ME model learned on the original (unclosed) training data.

Secondly, we must identify the temporal closure scoring function. For our experiments, this function takes into account the size of the temporal closure ($|T^\star|$) as well as the confidence values of the relations identified by the ME classifiers in the test set (not derived by the temporal closure algorithm) (only $\{c_{12}, c'_{23}, c'_{34}, c_{41}\}$ and $\{c"_{12}, c_{23}, c"_{34}, c"_{41}\}$ for the example shown in Figure 2). The correlation between these parameters and the scoring function is not straightforward. A preference for the confidence values favors closures which use only the top relations (in terms of confidence). However, weighing the size of the temporal closure leads to a result dominated by relations that close very well[10], such as *simultaneous* or *before* (which are also very frequent in the dataset). In the settings which produced the results shown in Table 7, we used the $score_1$ function: for $T = \{(R_1, c_1), \ldots, (R_k, c_k)\}$,

$$score_1(T^\star) = \lg(|T^\star|) \times \sum_{i=1}^{k} c_i.$$

The temporal relation accuracy drops by 3% after the relation selection among the top 3 best temporal relations for the testing documents. Posibile explanation: $score_1$ does not promote the close-to-gold temporal closures. The difinition of a good scoring function is not an easy process. Machine learning approaches might give us better coeficients for the parameters we consider. Alternativelly, our main assumption might prove incorrect:

[9] For each time-time pair, there is a single temporal relation with confidence equal to 1.

[10] When present, these relations will quickly generate many others in the temporal closure.

temporal closure is not a good indicator of a document's event ordering. The information conveyed by a document need not disclose a rich total ordering of its events.

5 Conclusion

In this paper, we briefly described our feature engineering efforts for temporal relation resolution and we analyzed three methods that exploit temporal reasoning and, more specifically, the closure of temporal relations, for the purpose of improving the performance of machine learned classifiers of temporal relations between events in text.

Based on our experiments, we find that feature engineering helps improve the classification problem, when compared with several baseline performances. However, given our current NLP capabilities, it is clear that we are faced with the performance bottleneck problem (accuracy below 60%). Any attempt to derive more advanced features demands more sophisticated methodologies of modeling temporal expressions, events and their relationships as well as advanced discourse understanding capabilities. For instance, the temporal duration or the start/end time points of events are highly useful for learning temporal relations. But, this introduces an even more challenging problem.

In terms of the utility of temporal reasoning in classifying temporal relation, the idea of using temporal reasoning to boost training data is certainly sound. But in order for the boosted training data to really take effect, more advanced features need to be investigated. Certainly, the process of dividing the data into training and testing has its impact on the system's performance and we are faced with the data sparseness problem. Temporal inconsistencies in our automatically labeled test dataset occurred in just a few test documents and the resolution process did not impact the system's performance. Improvements are needed in the process of selection of the to-be-replaced relations. Temporal data alignment largely depends on the function used to score the temporal closures and we plan to analyze the temporal closure of the training data and to explore other scoring functions.

References

Allen, J. F. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843.

Boguraev, B. and R. K. Ando. 2005. TimeML-Compliant Text Analysis for Temporal Reasoning. In *Proceedings of Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 997–1003, Edinburgh, Scotland, August.

Chambers, N., S. Wang, and D. Jurafsky. 2007. Classifying Temporal Relations Between Events. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 173–176, Prague, Czech Republic, June.

Chang, C. and C. Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `www.csie.ntu.edu.tw/˜cjlin/libsvm`.

Mani, I., M. Verhagen, B. Wellner, C. Min Lee, and J. Pustejovsky. 2006. Machine Learning of Temporal Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 753–760, Sydney, Australia, July.

Mani, I., B. Wellner, M. Verhagen, and J. Pustejovsky. 2007. Three Approaches to Learning TLINKs in TimeML. Technical Report CS-07-268, Computer Science Department, Brandeis University, Waltham, USA.

Min, Congmin, Munirathnam Srikanth, and Abraham Fowler. 2007. LCC-TE: A Hybrid Approach to Temporal Relation Identification in News Text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 219–222, Prague, Czech Republic, June.

Pustejovsky, J., P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. 2003. The TIME-BANK Corpus. In *Proceedings of Corpus Linguistics*, pages 647–656, Lancaster University (UK), March.

Tatu, Marta and Dan Moldovan. 2005. A Semantic Approach to Recognizing Textual Entailment. In *Proceedings of the HTL-EMNLP 2005*, pages 371–378, Vancouver, BC Canada, October.

Tatu, Marta and Dan Moldovan. 2007. COGEX at RTE3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 22–27, Prague, Czech Republic, June.

Verhagen, M., R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic, June.