Enhancing Temporal Relation Classification with MRS-based Features

William Lane
wlane@uw.edu

Abstract

We present an extension to ClearTK's TimeML submission to the 2013 TempEval shared task. The task involved the extraction of temporal relations from news texts, where a team used ClearTK's TimeML module to build the highest-scoring system (F1) for the temporal relation extraction task. We extended the intra-sentence event classification model of the ClearTK submission using semantic features derived from Minimal Recursion Semantics representations. Our results showed a modest improvement over the original ClearTK-based system, scoring 29.23% F1, which represents a .63% increase over the ClearTK-based system. The largest gain was in Recall, where our system scored 28.70%, representing a 2.1% increase over the original system.

1 Introduction

TempEval-3 was a shared task held in 2013 that focused on evaluating end-to-end temporal relation processing (UzZaman et al., 2013). The participants were given raw news article texts and were required to have their systems annotate that text for time relations between events. The ClearTK-TimeML system (Bethard, 2013) obtained the highest scores for relation extraction by using a relatively simple series of machine learning-trained models.

In our work, we take the system built by Bethard, and expand the feature set for one of the classification models to include Minimal Recursion Semantic representations (MRS; Copestake et al., 2005). An MRS is a semantic graph structure that links predicates (semantic nodes) to each other via predicate arguments (Copestake et al., 2005). For our work, the MRS was produced by parsing with the DELPH-IN consortium's Answer Constraint Engine (ACE) and the English Resource Grammar, which is a grammar that produces highly-precise syntactic constituent trees as well as its accompanying semantic representation for the sentence (ERG; Flickinger, 2000). Our system's feature extension extracts various predicate properties from the MRS, as well as crawls the path between events in the semantic graph to produce the features which contributed to the modest improvement in our results.

The rest of this paper will address in a bit more detail the background of the temporal relation extraction task, focusing particularly on the work done by Bethard on the ClearTK-TimeML system. I will then discuss the details of our feature design methodology, and the evaluation metrics we used to compare our system against others involved in the TempEval 2013 task. Finally, I will discuss the results of our system, and talk about possible areas of improvement.

2 Background and related Work

Given the complexity involved with producing an end-to-end temporal relation processing pipeline–which includes time extent and event detection in addition to temporal relation extraction–the TempEval 2013 shared task organizers decided to split the task into sub-tracks. Bethard approached the TempEval2013 temporal relation extraction sub-task by further breaking the problem down into sub-problems, then casting those problems as three seperate classification tasks for which he trained three models: one model to predict relations between events and time mentions in the same sentence, another to predict the temporal relation between the event and the document creation time, and a third model to predict the relation between same-sentence events. These models were then applied to predict the temporal relation label between pairs of events and/or times. The full set of possible relation labels in the TimeML inventory consist of: BEFORE, AFTER, INCLUDES, IS-INCLUDED, DURING, SIMULTANEOUS, IMMEDIATELY AFTER, IMMEDIATELY BEFORE, BEGINS, ENDS, BEGUN-BY, ENDED-BY, and NO-RELATION if the model was not confident in any other assignment.

For our purposes, we decided to focus our efforts on the event-to-event classification model,with the intuition that there exists more diverse and interesting semantic information between two event pairs rather than between event and timestamp pairs. This model came with a restriction on possible temporal relation labels. Bethard's system further reduced the relation extraction problem by imposing certain constraints on what kinds of events for which his system would attempt a prediction. For the inter-sentential model, in which we were most interested, the classifier relation domain was restricted to simply BEFORE and AFTER, with all other possible labels being lumped in with NO-RELATION. He also imposed a constraint on the possible syntactic paths that could exist between two events for which his system would attempt a prediction, by imposing the following constituent-and-movement pattern on all paris:
$((VP|ADJP|NP)?(VP|ADJP|S|SBAR)$
$((S|SBAR|PP))*((VP|ADJP)*|(NP)*)\$$

Finally, the features used by Bethard on the event-to-event classification model were as follows:

- Each event's class
- Each event's tense
- Each event's aspect
- The text of the first child of the grandparent of the event in a constituency tree
- The path through the constituency tree, from one event to the other
- The (unigram) tokens appearing between the two events

The above constraints and features make use of syntactic knowledge to attempt to maximize the system's prediction confidence. In addition to some of the more simple single-event features, the full syntactic path between events is explicitly considered both as a feature and as a constraint. The assumption here seems to be that if one can cluster event pairs with a similar path structure, that these inter-event paths must also by virtue of their similar structure contain some shared intrinsic semantic characteristic. The attempt seems to be using syntactic knowledge to grasp at semantic knowledge. This is where real semantic knowledge is needed, and so our system extension attempts to provide it.

3 Methodology

The framework provided by the ClearTK source code allowed us to write our own feature extractor class object which receives as input: two events, rendered as their surface string representation (usually a noun or a verb), their starting and ending indices, and the sentence they came from. Our method processes the input to produces a list of features which represents the relationship between the two events, and returns that list to the classification pipeline. The first features generated are the relatively low-hanging fruit: We extract the lemmatized predicate name of the event. Take for example the following sentence, which I will use from here on out to build a complete feature list:

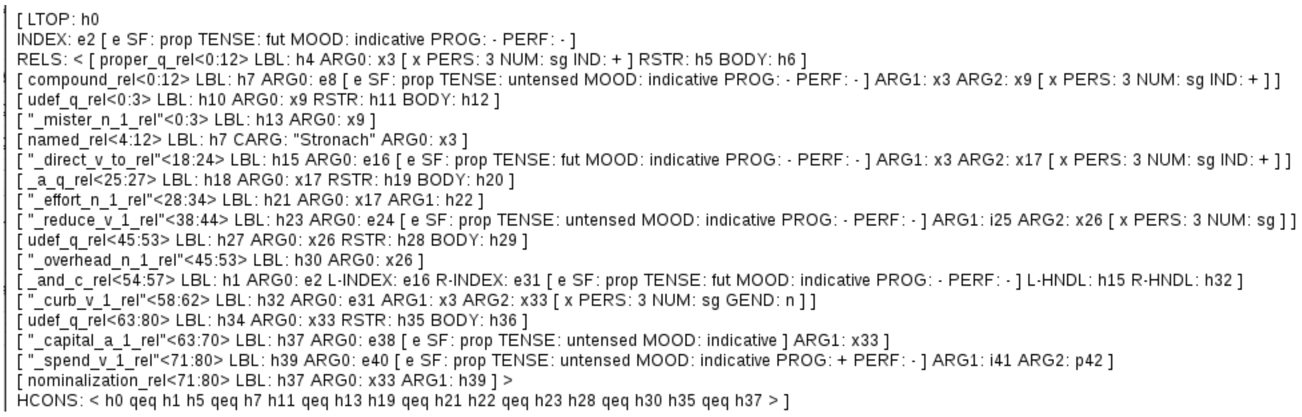
(1) *Mr. Stronach will direct an effort to reduce overhead and curb capital spending.*

Here, we designate event 1 (e1) as direct and event 2 (e2) as effort. The semantic representation produced by parsing this sentence with the ERG can be seen in figure 1.

A brief examination of the predicate list produced by this sentence allows us to locate the lemmatize forms of the two events we've just defined for the above example. In this particular case, the lemmatized forms are identical to the surface forms: e1=direct and e2=effort.

Second, we extract use information about the lexical item used to depict an event: If the event is a verb, we consider the mood and tense, as well as whether or not the aspect is progressive or perfective (if that information is available to us). If the event in a noun, we extract the attributes belonging to noun predicates: person, number, etc. Continuing with the same example sentence above, and appending these features to our current feature list, we get:

```
(2) e1=direct, e2=effort,
```



```
[ LTOP: h0
INDEX: e2 [ e SF: prop TENSE: fut MOOD: indicative PROG: - PERF: - ]
RELS: < [ proper_q_rel<0:12> LBL: h4 ARG0: x3 [ x PERS: 3 NUM: sg IND: + ] RSTR: h5 BODY: h6 ]
[ compound_rel<0:12> LBL: h7 ARG0: e8 [ e SF: prop TENSE: untensed MOOD: indicative PROG: - PERF: - ] ARG1: x3 ARG2: x9 [ x PERS: 3 NUM: sg IND: + ] ]
[ udef_q_rel<0:3> LBL: h10 ARG0: x9 RSTR: h11 BODY: h12 ]
[ "_mister_n_1_rel"<0:3> LBL: h13 ARG0: x9 ]
[ named_rel<4:12> LBL: h7 CARG: "Stronach" ARG0: x3 ]
[ "_direct_v_to_rel"<18:24> LBL: h15 ARG0: e16 [ e SF: prop TENSE: fut MOOD: indicative PROG: - PERF: - ] ARG1: x3 ARG2: x17 [ x PERS: 3 NUM: sg IND: + ] ]
[ _a_q_rel<25:27> LBL: h18 ARG0: x17 RSTR: h19 BODY: h20 ]
[ "_effort_n_1_rel"<28:34> LBL: h21 ARG0: x17 ARG1: h22 ]
[ "_reduce_v_1_rel"<38:44> LBL: h23 ARG0: e24 [ e SF: prop TENSE: untensed MOOD: indicative PROG: - PERF: - ] ARG1: i25 ARG2: x26 [ x PERS: 3 NUM: sg ] ]
[ udef_q_rel<45:53> LBL: h27 ARG0: x26 RSTR: h28 BODY: h29 ]
[ "_overhead_n_1_rel"<45:53> LBL: h30 ARG0: x26 ]
[ _and_c_rel<54:57> LBL: h1 ARG0: e2 L-INDEX: e16 R-INDEX: e31 [ e SF: prop TENSE: fut MOOD: indicative PROG: - PERF: - ] L-HNDL: h15 R-HNDL: h32 ]
[ "_curb_v_1_rel"<58:62> LBL: h32 ARG0: e31 ARG1: x3 ARG2: x33 [ x PERS: 3 NUM: sg GEND: n ] ]
[ udef_q_rel<63:80> LBL: h34 ARG0: x33 RSTR: h35 BODY: h36 ]
[ "_capital_a_1_rel"<63:70> LBL: h37 ARG0: e38 [ e SF: prop TENSE: untensed MOOD: indicative ] ARG1: x33 ]
[ "_spend_v_1_rel"<71:80> LBL: h39 ARG0: e40 [ e SF: prop TENSE: untensed MOOD: indicative PROG: + PERF: - ] ARG1: i41 ARG2: p42 ]
[ nominalization_rel<71:80> LBL: h37 ARG0: x33 ARG1: h39 ] >
HCONS: < h0 qeq h1 h5 qeq h7 h11 qeq h13 h19 qeq h21 h22 qeq h23 h28 qeq h30 h35 qeq h37 > ]
```

Figure 1: MRS representation of ”Mr. Stronach will direct an effort to reduce overhead and curb capital spending.”

`e1#MOOD#indicative, e1#PERF#-, e1#TENSE#fut, e1#PROG#-, e2#PERS#3, e2#IND#+, e2#NUM#sg`

The third feature class we extract is the shortest path between two events, determined by doing functor crawling from the second event sequentially to the first (regardless of which is the official e1 or e2) (Packard et al, 2013). If a path does not exist, we return NO_PATH. Again, building on the feature set for the example above, we see that the argument path through the MRS from e1 to e2 is relatively short: it is simply e1[verb] e2[noun], or as we notate it in our features:

`(3) e1v,ARG2#e2n`

In the general case, our event-to-event crawler follows the following algorithm:

1. Initialize variables to track the path string, as well as the recursion depth
2. In the MRS, start at the second of the two event's predicates:
   (a) Append “e2” to the path string, as well as the POS information for e2
   (b) Set e2's predicate to be the current predicate
3. For each predicate (P) that takes the current predicate (CP) as an argument:
   (a) If P is the predicate belonging to e1:
      i. Prepend “e1” to the path variable along with the POS information for e1.
      ii. *Return* the path string
   (b) Else:
      i. Prepend to the path string the argument position in which the current predicate appears.
      ii. Increment the recursion depth count, and if that count < 10 :
         A. Set P to be the new current predicate, and recursively dive back into step 3 (as long as the total recursion count has not exceeded 10)
      iii. Else:
         A. set path string to “NO_PATH”, and *return* the path string

In this manner, we are able to extract the relatively shorter paths between events, and if the actual full path is too long, or otherwise not reachable by simple functor crawling, we lump those cases together as having no path.

Additionally, there was about a 22% parse fail rate using ACE against the ERG. In these cases, features could not be extracted from any MRS representation, and so the system falls back on the default features produced by the original ClearTK-TimeML system.

4 Evaluation

Our system extension was evaluated against the original ClearTK-TimeML TempEval 2013 system using both the evlauation metric defined by Bethard and implemented internal to his system for the purpose of model-tuning, as well as the official TempEval-3 task evaluation metric distributed by the event organizers. The following sections discuss the two evaluation metrics and why we chose to report on both. We also explain the data on which the systems were trained and tested, as well as report on our comparative results.

4.1 ClearTK-TimeML Training Data

To run the Cleartk-TimeML system, we used the training data provided by the TempEval organizers. This included 183 TimeBank corpus documents, and 73 AQUAINT corpus documents all annotated for events, times, and relations in the TimeML format.

For test data, the TempEval-3 organizers provided a ”platinum” test data set for their final evaluation. For our development test data we had to pick a random subset of the given training data. Both the dev-test set and the platinum test set contain 20 documents.

Both systems were run on the same sets of training and test data, and finally evaluated on the same platinum corpus provided by the TempEval organizers.

4.2 TempEval Evaluation Metric

The 2013 TempEval shared task organizers provided an evaluation script so that the systems participating in the task could compare their results against each other. The temporal awareness evaluation metric uses precision, recall and f1 measures. The particulars of those measures are defined in figure 2.

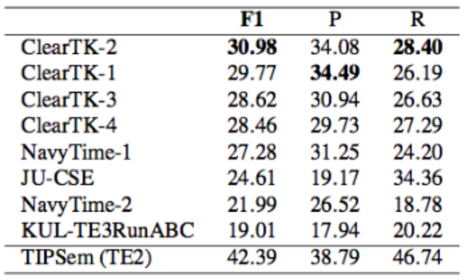
$$\boxed{\begin{aligned} Precision &= \frac{|Sys^{-}_{relation} \cap Ref^{+}_{relation}|}{|Sys^{-}_{relation}|} \\ Recall &= \frac{|Ref^{-}_{relation} \cap Sys^{+}_{relation}|}{|Ref^{-}_{relation}|} \end{aligned}}$$

Figure 2: Precision and recall for the temporal relation extraction task

What we see in figure 2 above is that precision is measured as the intersection of all the relations a system tags with the closure of the set of gold standard tags, all divided by the number of relations your system tagged. Basically, every tag you get right according to the gold standard, plus any tags you get right that are not explicitly labelled in the gold standard but could be logically inferred therefrom, divided by the number of labels your sytem produced.

Recall is measured as the intersection of the gold standard labels and the logical closure of the relations your system labeled, all divided by the number of relations labelled by the gold standard.

The evaluation setup was theoretically sound, and their allowance for giving credit for labels that could logically be inferred from the gold standard was very fair. However, there were some complaints raised by some of the participants in the task around the completeness of the provided training corpora. Steven Bethard in particular, the author of the TimeML module in ClearTK, included words on this subject in the discussion section of his system paper. He hypothesized, based on extensive error analysis which he had performed, that the AQUAINT corpus used in the test dataset for the official evaluation was not fully annotated. He observed that his system would mark relations that the annotators of the corpus has not spotted, relations which he judged to be correct. These cases would count against a total score assigned through the above-defined precision and recall methods, and as a result, every system in the TempEval2013 task recieved noteably lower scores than TIPSem, the provided state-of-the-art comparison system (see figure 3). For this reason, we decided to adopt additionaly the evlauation scheme that Bethard had built for the purpose of tuning models during the development phase of his original system.



	F1	P	R
ClearTK-2	**30.98**	34.08	**28.40**
ClearTK-1	29.77	**34.49**	26.19
ClearTK-3	28.62	30.94	26.63
ClearTK-4	28.46	29.73	27.29
NavyTime-1	27.28	31.25	24.20
JU-CSE	24.61	19.17	34.36
NavyTime-2	21.99	26.52	18.78
KUL-TE3RunABC	19.01	17.94	20.22
TIPSem (TE2)	42.39	38.79	46.74

Figure 3: TempEval 2013 end-to-end system track: Temporal awareness evaluations

4.3 ClearTK’s Internal Evaluation Metric

In Bethard’s internal evaluation setup, he takes the set of relation labels occuring in the intersection of the set of relations the system predicted, with the set of relations labeled by the gold standard. Precision, recall and F1 are defined formally as follows:

1. G=Gold system annotation pairs
   (a) For example: {(e1 BEFORE e2), (e2 AFTER e2), etc}

2. S=annotation pairs our system tagged

3. Precision = # correct relations $\in$ S / # relations $\in \{G \cap S\}$

4. Recall = # correct relations $\in$ S / # relations $\in \{G \cap S\}$

5. F1 = 2 * ((Precision * Recall) / (Precision + Recall))

Here we see that precision and recall are calculated the same way as the number of correctly-predicted system labels over the intersection of gold and system labels.

4.4 Results

We trained both the original ClearTk-based system and the ClearTk+MRS features system on the AQUAINT and Timebank corpora provided by the shared task organizers, and we tested both systems using the platinum corpus also provided by the task organizers. We first ran both systems, and examined the results of the internal evaluation metric, which provided us with a breakdown of precision and recall scores on a relation-type basis. In table 1, we can see that both systems scores the same F1 of .727 using the internal ClearTK-TimeML evaluation script. However, the total number of relations existing in the intersection of the gold and system relation label set has increased from 66 in the original system to 77 in our MRS-enhanced version, with a net gain of 8 more total correct labels in our system. This suggests that given a more standard scoring scheme–one that is not based on evaluation only of temporal relations occuring in both the gold and system relation set–recall scores would see a significant boost using our feature extension. Looking at the results produced by the official TempEval 2013 evaluation script confirms this:

Table 2: System results evaluated using the official 2013 TempEval task script

Original ClearTk			ClearTK + MRS		
F1	P	R	F1	P	R
28.53	29.53	27.59	29.23	29.79	28.70

Table 2 shows that our system improved on the original ClearTK-TimeML system, scoring 29.23% F1, which represents a .63% increase over the original system. The largest gain was in Recall, where our system scored 28.70%, representing a 2.1% increase over the original system.

5 Discussion

The results of our study are encouraging. We've shown that using very simple MRS features can improve a state-of-the-art temporal relation extraction system. While the overall gain in F1 was small, we believe that more could be accomplished in future iterations of this work. By adding more features into the SVM classifier, we inevitably introduce a certain amount of noise to the vector space. In this section, I discuss the conclusion we've drawn from error analysis on the final feature vectors as well as the final system classification output.

We examined the final feature vectors which contributed to the improved results, and found that the 22% parse failure rate was a significant source of noise. As explained in section 3, when an event-containing sentence fails to parse, the system simply defaults to to the original feature vectors generated by the ClearTK-TimeML system. This was happening in 22% of all training and testing sentences, which weakens potential patterns accross a number of dimensions in vector space. If some of those failed parses could be recovered, it would likely strengthen existing patterns found by the machine-learner, which could lead to significant improvement.

We examined many of the sentences that failed to parse and found that they are almost all very long, and full of multiple comma-delineated clauses which introduces more ambiguity and leads to an explosion of the space of possible parse trees. When this occurs and the number of possible parse trees exceeds a search space of 20000 passive edges, the parse fails, and our system is unable to extract MRS features from that sentence. In future work, one approach to improving ACE and the ERG's parse success rate could be to trim from the sentence some or all of the comma-delineated clauses which do not contain the event string, and see if the parser accepts the isolated fragment(s) which contain the relevent event information.

Additionally, we looked at the space of misclassified BEFORE and AFTER events for clues as to what types of errors were being made by our enhanced system. We produced a confusion ma-

Table 1: Results using ClearTK-TimeML system's internal evaluation script and the platinum test data set

ClearTk - TempEval 2013						ClearTK + MRS						
F1	P	R	#Gold	#System	#Correct	F1	P	R	#Gold	#System	#Correct	
.727	.727	.727	66	66	48	.727	.727	.727	77	77	56	OVERALL
.733	.815	.772	27	30	22	.774	.727	.750	33	31	24	AFTER
.722	.839	.776	31	36	26	.696	.889	.780	36	46	32	BEFORE
1.0	0.0	0.0	1	0	0	1.0	0.0	0.0	1	0	0	INCLUDES
0.0	0.0	0.0	0	0	0	1.0	0.0	0.0	1	0	0	IS_INCLUDED
1.0	0.0	0.0	7	0	0	1.0	0.0	0.0	1	0	0	SIMULTANEOUS

Table 3: Classification confusion matrix over the development test data

	AFTER	BEFORE	BEGUN_BY	ENDS	I-BEFORE	INCLUDES	IS-INCLUDED	SIMULTANEOUS
AFTER	27	5	0	0	0	0	0	0
BEFORE	6	59	0	0	0	0	0	0
BEGUN_BY	2	0	0	0	0	0	0	0
ENDS	1	0	0	0	0	0	0	0
I_BEFORE	0	1	0	0	0	0	0	0
INCLUDES	5	0	0	0	0	0	0	0
IS_INCLUDED	3	0	0	0	0	0	0	0
SIMULTANEOUS	6	6	0	0	0	0	0	0

trix based on the development test data to get an idea of where our errors were coming from. In Table 3, the labels along the first column represent the gold standard label, and the labels along the first row represent what our system classified the tempral relation as. The intersecting cells contain counts of occurences for each possible combination of system and gold standard label pairs.

Looking at the matrix, we immediately notice the presence of labels other than the BEFORE and AFTER with which we are primarily concerned in the event-to-event model. These represent other labeled temporal relations that exist in the training datasets, but for which Bethard did not attempt to classify due to their relative scarcity, and innate propensity for being ambiguous our difficult to annotate for in a consistent manner. However, these labels show up in the matrix because they exist in the gold data, and the system apparently attempted to label them as either BEFORE or AFTER. This accounted for 24 of the 35 total errors, according to Bethards evaluation on the dev-test data. The last 11 cases of error came from BEFORE and AFTER being confused for each other. The plethora of zeros in the majority of the columns is evidence of ClearTK's lack of effort to label relations besides BEFORE and AFTER in the inter-sentential model.

Beyond classifying the errors by building this confusion matrix, we also took a look at the BEFORE and AFTER labels and their mistakes between each other.We selected a sample of 11 misclassified event pairs and examined their sentenes. Of the 11 misapplied AFTER and BEFORE tags (instances of BEFORE classified as AFTER and vice-versa), 5 AFTER tags got misapplied to BEFORE relations, and 6 BEFORE tags got misapplied to AFTER relations. We categorized them into 4 classes of error based on observations we made on the actual sentences being classified:

1. Overlooking future, or conditional, or tentative constructions: 3 occurrences
   (a) We found that there were explicit semantic relationships between events, such as events being connected by modal or subjunctive constructions, that were not being picked up by the ClearTK system. Our features that we used in our extension did not explicitly target these constructions, but given that they are overtly semantic relations, it follows that they could be captured by deep semantic features.
2. Not using subordinate event tense as a reliable temporal cue: 2 occurrences
   (a) We noticed a couple cases where the tense of a verb event that was subordinate to another event would tell you when that event happened in relation to the first event.
3. Annotation Error: 2 occurrences

   (a) Some mistakes are just the gold standards fault.

4. NP events are difficult to place in time: 3 occurrences

   (a) A number of NP events were getting incorrectly classified. Our intuition tells us that in many situations where the NP event is the direct object of a verb event, then the NP event has likely already happened in relation to the verb event. ”He [saw] [the shooting]”, ”He [approves] of [the agreement]”. Of course, the interaction of the tense of the first event verb also plays a role in the temporal relation between the two events, so this heuristic may not hold unless you also consider tense and aspect of the verbal features as well. This is another avenue for more research that could be fruitful.

Out of the 11 misclassifications examined from the development testing data, only one of those specific samples was caught by our extension to the ClearTK system. It was one of the 3 instances from the first bucket enumerated above:

(4) *She e1=[said] the Miami relatives might [sue] for visitation*

In example 4, the MRS path feature corresponding to the argument path between event 1 (said) and event 2 (sue) captured the unique semantic structure of the predicates. The predicate attached to ”sue” is subordinate to the predicate attached to ”said” through the second argument position followed by a first argument position of another verbal predicate. The corresponding path feature looked like example 5:

(5) e1#v,ARG2#v,ARG1#e2#v

In the SVM, this example would have been plotted in vector space close, at least in this single dimension, to other event pairs with an identical semantic argument structure. Over enough iterations and occurences of this feature in the data set, the machine learns that the semantic structure where the second event is a verb and also the direct argument of another verb which in turn is subordinate to a first event (which is also a verb), is indicative of the first event coming BEFORE the second event,temporally speaking. This of course should not be taken as a hard and fast rule of linguistics, but it is the kind of subtle heuristic that machine learning can leverage when you use deep semantic features like paths through an MRS.

6 Conclusion

In this study we approached the task of temporal relation extraction by expanding the feature set of ClearTK’s TimeML-based system, which was the top-scoring system in the 2013 TempEval shared task. Our focus was to determine whether the addition of Minimal Recursion Semantic representations could boost the performance of a complete temporal relation extraction system. Our additions to the base feature set included: lemmatized renderings of the MRS predicates correlating with the given events, the set of verbal attributes produced by events in the MRS representation, and the shortest path between events by performing functor crawling through an MRS. We found that our semantic features resulted in a .63% increase in F1 over the ClearTK-based system. The largest gain was in recall, where our system scored 28.70%, representing a 2.1% increase over the original system. Looking forward, we see oppurtunities to improve our efforts here by doing more effective sentence simplification to improve the parse success rate of the ACE and the ERG. We also see oppurtunity to craft further semantic features around explicit temporal predicate cues, as well as experiment with different MRS crawling techniques to capture different kinds of relationships.

7 Acknowledgements

Thanks to professor Emily Bender first of all, for her direction and assistance in relevant matters both theoretical and pragmatic.Thanks as well to Martin Horn, who implemented the feature extraction algorithms that we together designed.

References

Steven Bethard. 2013. ClearTK-TimeML: A Minimalist Approach to TempEval2013. *Second Joint Conference on Lexical and Computational Semantics, Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation*, SemEval-2013, Atlanta, Georgia. Association for Computational Linguistics, n.d.Web<http://www.aclweb.org/anthology/S13-2002>.

Branimir Bogurev et al *TimeML 1.2.1: A Formal Specification Language of Events and*

Temporal Expressions. October, 2005. Web. <http://www.timeml.org/publications/ timeML-docs/timeml_1.2.1.html>

Naushad UzZaman et al SemEval-2013 Task 1: TEMPEVAL-3: Evaluating Time Expressions, Events and Temporal Relations. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), pg 1-9. Atlanta, Georgia, June 14-15, 2013.* Association for Computational Linguistics, June

Copestake, D. Flickinger, C. Pollard, and I. A. Sag 2005. Minimal Recursion Semantics: An Introduction. *Research on Natural Language and Computation,*, 3(4),pp. 281 -332.

D. Flickinger. 2000. On building a more efficient grammar by exploiting types *Natural Language Engineering*, 6(1),pp. 15-28.

W. Packard, E. M. Bender, J. Read, S. Oepen and R. Dridan. 2014. Simple Negation Scope Resolution Through Deep Parsing: A Semantic Solution to a Semantic Problem. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.*, Baltimore, MD.