Enhancing Temporal Relation Classification with MRS-based Features

Martin Horn
Department of Linguistics
University of Washington
Seattle, WA 98195
mrtnhorn@uw.edu

Abstract

This project explores the use of semantic features in improving a system which extracts and relates temporal expressions and events in general domain text. We use a top-performing system which participated in TempEval-3 as a baseline and add features derived from Minimal Recursion Semantics (MRS) representations derived from parsing the shared task data. Specifically, we focus on the component of the system which classifies the relation between two intra-sentential events. We have found that incorporating the MRS into the existing feature set improves F1 score by 2.5% using official shared task test data and evaluation metric.

1 Introduction

Identifying and relating temporal expressions in free text remains a difficult and typically low-performing task in NLP. There are many reasons for this: temporal expressions vary widely in form, they are difficult to normalize, and classifying their relation to each other requires context. However, the ability to perform these tasks is very useful for a variety of domains. For example, in the clinical domain, interpreting temporal information in clinical notes can give valuable insight into patient symptoms, diagnoses, and treatments.

This task is so difficult yet important that an entire temporal shared task has emerged within the umbrella of SemEval. In this paper, we work on the TempEval-3 task from 2013, which focuses on "the automatic identification of temporal expressions (timexes), events, and temporal relations within a text" (UzZaman et al., 2013).

Existing solutions to this task have typically been either rule-based systems or machine learning systems which incorporate n-gram and/or paths from syntactic constituency or dependency trees. We experiment using features derived from semantic representations in order to improve temporal relation classification, which is inherently a semantic task. We use Minimal Recursion Semantics (MRS) analyses (Copestake et al., 2005), from which we derive features to be added to a baseline system which uses an SVM classifier. To our knowledge, this is the first time that features derived from MRS analyses have been used in a temporal information extraction task.

We have found that adding these semantic features to an existing baseline feature set of n-grams, syntactic paths, and morphosyntactic properties increases F1 score by as much as 2.5% and recall by as much as 4%. In Section 2, we cover some related work in deriving MRS features, the target task we are approaching, and the baseline system we build upon. We outline our main approach in Section 3, provide our evaluation metric and results in Section 4, and finally discuss our results and perform error analysis in Section 5.

2 Related Work

MRS has been used in a variety of NLP tasks. Packard et al. (2014) crawl MRS analyses in order to perform a rule-based approach to negation scope resolution. Kramer and Gordon (2014) use similar crawling techniques in order to design features for use in a Naive Bayes-based sentiment classification system. In this work, we look at expanding this use of MRS crawling to a new domain where semantic dependency paths are very relevant.

2.1 Target Task

The target task that we are approaching is the Task C component of TempEval-3 (UzZaman et al., 2013).

Task C involves identifying and classifying pairs of entities with a temporal link (TLINK). Entities can be either events or timexes, and the set of TLINK values are {BEFORE, AFTER, INCLUDES, IS-INCLUDED, DURING, SIMULTANEOUS, IMMEDIATELY AFTER, IMMEDIATELY BEFORE, IDENTITY, BEGINS, ENDS, BEGUN-BY, ENDED-BY}. However, the creator of our baseline system restricts the system classifier relations to {BEFORE, AFTER}, most likely due to sparsity in the other relations (Bethard, 2013).

Teams that wished to only complete Task C could either identify the pairs of entities that have a TLINK and classify those TLINKS (given the the gold entities, but not which entities were related) or just classify the TLINKS (given the gold entities as well as the related pairs) (UzZaman et al., 2013). Our baseline both identifies and classifies the TLINKS, but we will be working on the classification component. Specifically, we are adding features to the component that classifies relations between two events in the same sentence.

2.2 Baseline System

The baseline system that we are using is called ClearTK-TimeML. It was the best performing system (by F1 score) in the TempEval-3 end-to-end temporal relation extraction task (Task ABC). ClearTK is a machine learning and NLP framework for information extraction developed by the University of Colorado's Center for Computational Language and Education Research group. The Time-ML module was developed by Steven Bethard, now at the University of Alabama (Bethard, 2014).

Though the system performs temporal entity extraction in addition to relation identification/classification, the focus here will be on the relation component since that is the most intuitively relevant for MRS incorporation. An example of the gold standard annotation for a sentence from the training data can be found in the Appendix. This annotation includes a marked up version of the sentence with temporal expression and event entities extracted, the instances created for two of the events, and two example TLINKs.

In the [Event] to [Event] model, only pairs whose syntactic path from one event to the other matched the following regex of syntactic categories and up/down movements were considered:

$$\hat{}((VP\uparrow|ADJP\uparrow|NP\uparrow)?$$
$$(VP|ADJP|S|SBAR)(\downarrow(PP|SBAR|S))*$$
$$((\downarrow VP|\downarrow ADJP)*|(\downarrow NP)*)\$.$$

The domain of labels was {BEFORE, AFTER}, and the features used were (Bethard, 2013):

- Each event's aspect, class, and tense
- The text of the first child of the grandparent of each event in the constituency tree
- The path through the syntactic constituency tree from one event to the other
- The tokens between the two events.

In our approach, we also incorporate property and n-gram-like features; however, our main contribution is the use of semantic dependency paths, which can capture more direct and relevant relationships between events than paths through a constituency tree.

The data for this shared task includes three separate corpora. The normal gold corpus consists of human-annotated data from TimeBank (61,418 words) and AQUAINT (33,973 words). An additional "silver" corpus was provided for additional training. This corpus was created automatically on the Gigaword corpus (666,309 words) by merging three state-of-the-art systems from previous shared tasks (UzZaman et al., 2013).

Bethard tested a few different machine learning classifiers but used LIBLINEAR SVM and logistic regression for all of the final models. The data used to train the models came from the task-provided TimeBank, AQUAINT, and silver annotated corpora, as well as a corpus of verb-clause temporal relation annotations from Bethard et al. (2007) and closure-inferred temporal relations from Muller (2013). The best performing models used TimeBank and Bethard's own corpus. We did not have access to Bethard's own corpus, so we trained on TimeBank and AQUAINT.

3 Methodology

Our general approach is threefold: parse the training and test sentences in order to obtain MRS representations, process the MRS in order to extract features, and inject those features into the baseline's feature set. The hope is that given a sentence and two events which have been pre-identified from that sentence, our semantic features will improve the machine learner's ability to classify the

[LTOP: h0 INDEX: e2 [e SF: prop TENSE: past MOOD: indicative PROG: - PERF: -] RELS: <[pron<0:4>LBL: h4 ARG0: x3 [x PERS: 3 NUM: pl PT: std]] [pronoun_q<0:4>LBL: h5 ARG0: x3 RSTR: h6 BODY: h7] [_yell_v_1<5:11>LBL: h8 ARG0: e9 [e SF: prop TENSE: past MOOD: indicative PROG: - PERF: -] ARG1: x3 ARG2: p10] [_and_c<12:15>LBL: h1 ARG0: e2 L-INDEX: e9 R-INDEX: e11 [e SF: prop TENSE: past MOOD: indicative PROG: - PERF: -] L-HNDL: h8 R-HNDL: h12] [_weep_v_1<16:20>LBL: h13 ARG0: e11 ARG1: x3 ARG2: p14] [_when_x_subord<21:25>LBL: h12 ARG0: e15 [e SF: prop TENSE: untensed MOOD: indicative PROG: - PERF: -] ARG1: h16 ARG2: h17] [_the_q<26:29>LBL: h18 ARG0: x19 [x PERS: 3 NUM: sg] RSTR: h20 BODY: h21] [**_decision_n_1**<30:38>LBL: h22 ARG0: x19] [**_announce_v_to**<43:53>LBL: h23 ARG0: e24 [e SF: prop TENSE: past MOOD: indicative PROG: - PERF: -] ARG1: i25 ARG2: x19 ARG3: i26] [parg_d<43:53>LBL: h23 ARG0: e27 [e SF: prop TENSE: untensed MOOD: indicative PROG: - PERF: -] ARG1: e24 ARG2: x19] >HCONS: <h0 qeq h1 h6 qeq h4 h16 qeq h13 h17 qeq h23 h20 qeq h22 >]

Figure 1: MRS Analysis for *They yelled and wept when the [decision] was [announced].*

relation between the events as BEFORE or AFTER.

3.1 ERG Parsing

In order to attain MRS analyses, we use an exhaustive hand-built precision grammar called the LingGO English Resource Grammar (ERG)[1] (Flickinger, 2000). We use the Answer Constraint Engine (ACE)[2] in order to parse a given sentence with the ERG and receive the MRS analysis for the sentence. Because the ERG is a hand-built precision grammar, it sometimes cannot parse a given sentence, whether that be due to an ungrammatical structure, misspelling, or too lengthy of a sentence. Over all of our data (training, dev, and test), 21.8% of our sentences return no parse. In the case of a sentence with no parse, we do not add any additional features and treat the baseline feature set as a back-off. Figure 1 shows an example of an MRS analysis from the ERG for the sentence *They yelled and wept when the decision was announced.*

3.2 Feature Design

Once we have an MRS analysis for a given sentence, we connect the two given events to elementary predications (EPs) in the MRS using character spans. In Figure 1, the noun *decision* and the verb *announced* have been selected as the two events in the sentence. We then process the MRS analysis using pyDelphin[3] for both static features (explicitly given strings in the MRS) or those attained from crawling the MRS graph.

[1]http://www.delph-in.net/erg/

[2]http://sweaglesw.org/linguistics/ace/

[3]https://github.com/goodmami/pydelphin

3.2.1 Static Features

For each event, we add its corresponding EP's lemma name, such as `e1LEMMA=decision` and `e2LEMMA=announce` for the events in Figure 1. These normalized lemmas may be useful as features because they add a level of abstraction from the surface string. Morphological variation in words can lead to data sparsity, so using lemmas may capture broader patterns.

For each event, we also include the properties of the associated EP. For example, for the events in Figure 1, we create the following feature set:

- `e1NUM=sg`
- `e1PERS=3`
- `e2TENSE=past`
- `e2PROG=-`
- `e2SF=prop`
- `e2MOOD=indicative`
- `e2PERF=-`

Some of these properties, especially the verb properties, can be useful cues for determining when one event occurred in relation to another. The baseline system already includes tense and aspect features, but we are experimenting adding them as well to see if they may reinforce the existing features (if the features assign the correct value) or counteract them (acting as "good noise" if the existing values are incorrect). Mood may be useful because subjunctive verbs refer to events that likely have not occurred.

3.2.2 MRS Crawling

Beyond using static labels explicitly stated in the MRS representation, MRS can also be traversed

by using the analysis as a semantic dependency graph. Each EP has a set of arguments which connect the EP to other EPs. We hope to capture semantic relationships by finding a path from one event to another in a sentence. We follow Packard et al. (2014)'s approach of functor crawling for our *full path* feature and argument crawling for our *direct path* feature.

First, we find the shortest path between the two event EPs, starting at the second-occurring event in the sentence and functor crawling back until we reach the first event. The idea here is that we want to capture semantic relations between events, rather than just a syntactic path as is used in the baseline system. We include the event number as well as part of speech and argument number for each node in the path. For example, the feature for the path between e1 (*unwilling*) and e2 (*make*) in (1) is `PATH=e1#a,ARG2#e2#v`.

(1) *Publicly, they are [unwilling] to [make] any sweeping assertion.*

The MRS in Figure 1 returns the feature `PATH=NO_PATH` because the events are not connected through functor crawling.

We also use argument crawling to find situations in which one event is in the argument set of the other event. These provide us with "direct" paths between the two events. It is possible for this feature to look very similar to the PATH feature above. This is the case in (1), for which our direct path feature is `DIRECT=e1#a[ARG2#e2#v]`.

However, in cases like the sentence in Figure 1, where no functor path is found, the argument-crawled path comes in handy:

`DIRECT=e2#v[#ARG2#e1#n]`.

Here, the MRS captures the passive nature of the verb *announced* and the fact that *decision* is its argument rather than vice versa (as the order in the sentence might falsely indicate just looking at n-grams). If this happens enough, the machine learning system may pick up the pattern that a noun which occurs as the ARG2 of a verb tends to happen before the verb. This is the case for the noun *decision* in relation to the verb *announced* in Figure 1.

3.3 Incorporating Features into Baseline

In order to use these new features in the baseline system, we add a module to the existing feature extraction pipeline. This pipeline provides a sentence and the two extracted events from that sentence which must be related. Our additional module generates the new feature sets based on this sentence and pair of events and sends back a feature list. These features are then converted into UIMA feature objects, which ClearTK then automatically adds into the final feature vector representing the event pair and sentence, along with the baseline features.

4 Evaluation

In order to test our feature additions to the system for development purposes, we used a random selection of documents from the training set as devtest data. We used an internal ClearTK evaluation script to test this development data. Once we finished tuning our system, we tested on the held-out official shared task evaluation test data, using both the internal evaluation script and the official shared task evaluation script.

4.1 Datasets

Our development test data is a random of 20 documents from the TimeBank and AQUAINT corpora. Our evaluation test data consists of 20 documents (6,375 words) from the shared task's "platinum" corpus (also taken from AQUAINT), which was annotated by the task organizers and received a high inter-annotator agreement (UzZaman et al., 2013).

4.2 Metrics

Evaluation for temporal relation processing for the original shared task incorporated the idea of *temporal awareness*: "how well pairs of entities are identified, how well relations are categorized, and how well the events and timexes are extracted" (UzZaman et al., 2013). In this measure, explicitly mentioned relations are used (*reduced relations*), as well as implicit relations inferred via closure (*temporal closure graph*) using a technique from Muller (2013). Precision was calculated as the number of reduced system relations that can be verified from the reference annotation temporal closure graph out of the number of temporal relations in the reduced system relations. Recall was the number of reduced reference annotation rela-

ClearTK Baseline						ClearTK Baseline+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 1: Internal Evaluation on Platinum Data

tions that can be verified from the system output's temporal closure graph out of the number of temporal relations in the reduced reference annotation (UzZaman et al., 2013).

In addition to using the official TempEval-3 evaluation script on our test data, we use the evaluation code included with ClearTK-TimeML (which was also used for development purposes). This code implements a modified evaluation method, due to the creator's disappointment with the TempEval-3 measure (which in turn stemmed from reportedly incomplete annotations[4]) (Bethard, 2013).

The internal measure only looks at the intersection of the reference annotations and the system annotations (ignoring, for instance, system annotations that don't appear in the reference). Essentially, this means that overall precision, recall, and F1 for all three types of relations are both more of a measure of accuracy, and thus identical in value. That is, they are the number of correct system annotations (class labels for relations are identical) out of the intersection of the system and reference annotations (the relations that exist between system and reference, whether or not they have the correct labels). So, both precision and recall are calculated as 56 correct out of 77 total gold/system relations for our system performance in Table 1.

The precision and recall for each relation type (for example, just AFTER or just BEFORE) are more typical metrics. Precision in this case is the number of correct system relations over the total number of system relations, while recall is the number of correct system relations over the total number of gold relations. F1 score is standard: $2 \times (P \times R)/(P + R)$.

4.3 Results

The results of running our trained system on the 20 platinum documents using the aforementioned internal evaluation script are shown in Table 1, while the results using the official script are shown in Table 2. With the internal evaluation script, our F1, precision, and recall scores remain the same with our added features. However, we include the actual gold, system, and correct counts to show that though the scores have stayed the same, the total sentence/event pair combinations evaluated has changed from 66 to 77 (this will be discussed in Section 5).

Table 2 shows that our added features have increased recall by 1.11, or a 4% increase over the baseline. Precision has also gone up slightly, resulting in an F1 score increase of 0.7, or 2.5%, over the baseline.

ClearTK Baseline			ClearTK Baseline+MRS		
F1	P	R	F1	P	R
28.53	29.53	27.59	**29.23**	**29.79**	**28.70**

Table 2: Official Evaluation on Platinum Data

5 Discussion

A cursory glance at the internal evaluation results in Table 1 shows that adding MRS-based features didn't improve overall precision, recall, and F1. However, as described in Section 4.2, this metric only looks at the instances in which the gold standard and system both provide TLINK for the given event pair. Changing the feature set can actually change which instances are given a TLINK by the system, and thus change the intersection between gold and system predictions. So, while our actual *score* didn't change, the number of event pairs judged by the system increased by 11. Theoretically this should improve recall, but this internal metric is in a sense *recall-blind*.

This means that our improvements are best seen in the official evaluation script, which is *recall-sensitive*. While the platinum annotations may have been incomplete, which can hurt precision if a system annotates a relation that isn't in the plat-

[4]ClearTK was annotating apparently valid relations which didn't exist in the gold annotations, causing a hit in precision.

inum, the official evaluation metric allows recall to improve because it includes the symmetric difference of the platinum and system annotation sets. Since our feature additions cause the system to annotate more relations than it was in the baseline, recall sees the largest gain. We surmise that this may be due to the fact that the extra features allow the baseline system to cross a confidence threshold regarding a relation that it wasn't able to with the baseline feature set.

5.1 Feature Combinations

We experimented with a variety of feature combinations and found that the success of different combinations was not always predictable given the performance of a feature addition on its own. For example, adding just the lemma feature to the baseline actually decreased scores, while adding it to properties and paths was more effective than properties and paths alone. This is to be expected, as the co-occurrence of certain features may be more effective than the features alone.

We also experimented with features that were not used in our final results. One type of feature in particular that we expected to boost scores was at looking for a limited set of EPs (or EP substrings) which are particularly informative to the temporal relation between events: {*_modal, *_temp_loc, _when_x_subord, _once_x_subord, _after*, _during*, _while*, _before*, _until* _as_x_subord, _since_x_subord}.

After identifying an EP containing one of the above strings, we argument crawled through the graph, beginning with the given EP's arguments. If the EP was connected to both events via arguments, we then add a feature such as PRED=e1#_after_p#e2. We hoped these features would capture direct semantic relations missed by an n-gram/syntactic approach, possibly even with an EP that directly corresponds to the desired relation label (such as AFTER in this case). However, these features were too sparse in order to capture patterns in the data—a mere 98 occurrences out of 5,980 training/devtest instances.

We also experimented with a method of automatically extracting predications like those above by finding any predication which was connected to both events via argument crawling and creating an analogous feature. While this greatly increased the number of these features added (812 on training/devtest instances), it still did not improve scores. The reason for this may be that many of the added predications were not useful and led to increased noise. For example, many of these "important predicates" were parg_d, which is likely not useful information for temporally relating two events. In the future, it may be useful to implement some sort of filtering so that there is a compromise between number and relevance of features.

5.2 Error Analysis

For development purposes, we looked at 10 classification errors of BEFORE and AFTER relations in the baseline event-to-event model. Since these were the relations that the baseline system predicted, they were what we were hoping to improve. We classified the errors into 5 categories, based on what seemed to be natural groups of grammar-related errors:

1. Overlooking future, conditional, and "tentative" constructions (3 errors)
2. Not using subordinate verb phrase tense (2 errors)
3. Annotation error (2 errors)
4. Misinterpreting NP events (3 errors)

Type 1 errors involved a construction in which the system did not pick up on the tense, mood, or tentativeness of the relations (which were determined by syntax), and instead incorrectly made a judgment (seemingly) based on morphology. For example, in sentence (2), the modal *might* creates a conditional mood which implies that *sue* hasn't happened yet and therefore happens after *said*. For some reason, the baseline system misclassified *said* as AFTER *sue*.

(2) *Elian's great-uncle vowed to continue the battle, and [said] the Miami relatives might [sue] for visitation while the child is in the United States.*

Type 2 errors were judged based on our hypothesis that the tense of the verb in a subordinate clause should be used to determine its relation to

	AFTER	BEFORE	BEGUN_BY	ENDS	IBEFORE	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
IBEFORE	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

Table 3: Confusion Matrix on Devtest Data for ClearTK Baseline+MRS

the outer verb. In these instances, the system did not take this idea into account. For example, in (3), *dispatched* occurs inside of the constituent headed by the verb *criticized* and occurs beforehand temporally. The baseline misclassified *criticized* as BEFORE *dispatched*.

(3) *The move seemed aimed at heading off more trouble with Iran, which had condemned Iraq's invasion of Kuwait on Aug. 2 but also [criticized] the multinational force [dispatched] to Saudi Arabia.*

Type 3 errors were caused by seemingly incorrect annotations, where the system actually got the right classification. For example, the baseline "misclassified" *preparing* as BEFORE *vote*, which seems like the correct classification.

(4) *Already one grandstanding Indiana Republican has subpoenaed Elian to testify before a House panel, and others in the Senate are [preparing] to [vote] the boy U.S. citizenship.*

Type 4 errors involved a noun phrase as one of the events, and the relation of the NP to the other event was misinterpreted. For example, in (5), the noun *ruling* occurs temporally before *called*, but the baseline misclassified *called* as BEFORE *ruling* (most likely due to the past tense of the verb).

(5) *Jose Basulto, founder of Brothers to the Rescue, a Cuban exile group that frequently rescues Cuban refugees from the Straits of Florida, [called] the latest [ruling] outrageous.*

The confusion matrix in Table 3 shows our system (columns) vs. gold (rows) performance on our devtest data. It is evident that our system only uses the AFTER and BEFORE labels, but that the number of other labels is quite small. 35 out of the 121 devtest instances were misclassified, most of which were misclassified as AFTER. Out of the 10 classification errors observed above, two of them, (2) and (3), were fixed with our addition of MRS-based features.

In (2), our improved system correctly classified *said* as BEFORE *sue*. This was able to happen without a feature that explicitly connected the events with *might*, despite this example being one of our motivations for experimenting with the unhelpful unique predication feature. In (3), our improved system correctly classified *criticized* as AFTER *dispatched*, which may be due to the ERG's effective handling of passive forms (as was shown in Figure 1).

While it would be nice to see more of these misclassifications get fixed, it is promising to see both a Type 1 and Type 2 error corrected, resulting in a 20% error reduction on our error analysis selection. We were hoping that some of the noun event errors would be fixed, especially due to the two different path features which specify part of speech, but none of our selected devtest instances were corrected.

6 Conclusion

Our motivation in this project was to apply semantic representations to an inherently semantic task. Task C of TempEval-3 involves classifying the temporal relationship between events in the same sentence, which is a task which could greatly benefit from direct semantic dependencies, rather than just n-gram and syntactic features. We include MRS-based features into an existing SVM classifier baseline, which results in modest but definitive improvements using the shared task's official evaluation metric. We show through error analysis what features did and did not prove effective, as

well as some concrete examples of improvements that our additions made.

In the future, we would like to do more rigorous testing of all of the different feature combinations, as well as fine-tuning existing features. For example, initial testing on narrowing down the property feature set showed promising results and could be more thoroughly fleshed out. Additionally, we would like to expand from just classifying intra-sentential event relations to including temporal expression relations and maybe even extracting these temporal entities as well.

Acknowledgements

My partner on this project was William Lane. Due to his Java experience and the fact that he found the baseline system that ended up working for us, he did most of the actual coding and configuring on the baseline system. We parsed the sentences and worked with MRS analyses using pyDelphin; since I feel comfortable in Python, I did most of the MRS crawling and feature engineering, while Will incorporated the new features into the existing system.

References

Steven Bethard, James H. Martin, and Sara Klingenstein. 2007. Finding temporal structure in text: Machine learning of syntactic temporal relations. *International Journal of Semantic Computing*.

Steven Bethard. 2013. ClearTK-TimeML: A minimalist approach to TempEval 2013. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10-14. Atlanta, Georgia, June 14-15, 2013. Association for Computational Linguistics.

Steven Bethard, Philip Ogren, and Lee Becker. 2014. ClearTK 2.0: Design Patterns for Machine Learning in UIMA. *LREC 2014*. Reykjavik, Iceland. European Language Resources Association (ELRA). <http://www.lrec-conf.org/proceedings/lrec2014/pdf/218_Paper.pdf>.

Branimir Boguraev, et al. 2005. *TimeML 1.2.1: A Formal Specification Language for Events and Temporal Expressions*. Oct. 2005. <http://www.timeml.org/publications/timeMLdocs/timeml_1.2.1.html#timex3>.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *R*esearch on Language and Computation, 3(4), 281-332.

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

Jared Kramer and Clara Gordon. 2014. Improvement of a Naive Bayes Sentiment Classifier Using MRS-Based Features. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 22-29. Dublin, Ireland.

Philippe Muller. 2013. Closure Temporal Relation Inferences. <https://groups.google.com/d/topic/tempeval/LJNQKwYHgL8>.

Woodley Packard, Emily M. Bender, Jonathon Read, Stephan Oepen, and Rebecca 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.

Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen, and James Pustejovsky. 2013. 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)*, pages 1-9. Atlanta, Georgia, June 14-15, 2013. Association for Computational Linguistics.

Appendix

Original sentence

In Washington today, the Federal Aviation Administration released air traffic control tapes from the night the TWA Flight eight hundred went down.

Sentence with TIMEX and EVENT entities extracted

In Washington <TIMEX3 tid="t53" type="DATE" value="1998-01-14" temporalFunction="true" functionInDocument="NONE" anchorTimeID="t0">**today**</TIMEX3>, the Federal Aviation Administration <EVENT eid="e1" class="OCCURRENCE">**released**</EVENT> air traffic control tapes from the night the TWA Flight eight hundred <EVENT eid="e2" class="OCCURRENCE">**went**</EVENT> down.

Instances created from events

<MAKEINSTANCE eventID="e1" eiid="ei252" tense="PAST" aspect="NONE" polarity="POS" pos="VERB" />

<MAKEINSTANCE eventID="e2" eiid="ei253" tense="PAST" aspect="NONE" polarity="POS" pos="VERB" />

TLINK identification/classification

(My summary of relation parentheses)

<TLINK lid="l2" relType="INCLUDES" timeID="t53" relatedToEventInstance="ei252" />

(*today* INCLUDES *released*)

<TLINK lid="l5" relType="AFTER" eventInstanceID="ei252" relatedToEventInstance="ei253" />

(*released* AFTER *went*)