Online Appendix A

Features for the classifiers

The feature engineering efforts in this task built upon our previous work, notably the relation models[5] which achieved top performance in the (non-temporal) semantic relation task of the 2010 i2b2 NLP challenge. While we explored a wide variety of features, motivated from different perspectives, we report only those that consistently improved performance in ten-fold cross-validation.□

Local Link Specialist features

Surface features describe how two extents are superficially linked within the text. These include word/extent n-grams, word/extent sequences, discretized counts of words, punctuation marks, extents, features that capture the order of extents, and conjunctions. Table A1 summarizes the surface features. In general, we replaced words with extents where possible, and introduced hybrid n-grams, which mix words with extents. As an example: given that ‘multiple angioplasty stents’ was known to be an *event* of type *Treatment,* the phrase “post multiple angioplasty stents” would be processed as a bigram “post <event::treatment>” rather than as a word 4-gram. The other attributes of an extent, including *modality* and *polarity*, were also considered but the *type* attribute was the most effective one. All our features are binary; therefore, non-binary features such as counts are binned. For example, if the number of words between extents is between 10 and 25, a binary ‘extents-far-apart’ feature is generated.

Clinical narrative often lists events in conjunction, e.g. “the patient experienced episode of tongue biting, unresponsiveness, disorientation“. Therefore we decided to handle them further by introducing Conjunction features to signal cases where the tokens between the two extents are only conjunctions and extents. Specifically, we found that if we skip all conjunction coordinates as well as extents, when extracting before-, after-, and between-extent ngrams, we got better results. As an example, when we need to decide the relation between e1 and e4 in the sentence "e1 was admitted to the hospital after the patient felt e2, e3, and e4", the four-word-before features for e4 are better to be {after, the, patient, felt} than {felt, e2, ,, e3}. This strategy was applied to all applicable features, including those involving POS tags discussed below.

Table A1: Surface features describing the text between and around E1&E2 (the two local extents under consideration).

Hybrid ngrams:	using the values of the extent’s ctype (Event, Time, or Sectime) and type (Problem, Test, Date, etc). - four words/extents before and after each of E1&E2; - words/extents in between E1&E2, word bigrams between E1&E2; - words within E1&E2, word bigrams within E1&E2; - attribute pairs between E1&E2 (using all attributes, not only type and ctype).
Sequences:	- for sequences shorter than a threshold length, the feature is enhanced with the type and ctype values of E1&E2 to differentiate, e.g., "<Treatment> for <Problem>" is typically an ‘After’ relation, whereas "<Treatment> for <Duration>" is typically an ‘Overlap’ relation.
Counts:	- bigrams before and after each of E1&E2; - word/extent sequence between E1&E2; - numbers of words, punctuations, extents in between E1&E2.
Orders:	- for both event-event and event-time relations, indicating the ctype of the first extent.
Conjunctions:	- indicating if the tokens between E1&E2 are only conjunctions and extents; enhanced with ctype and type.

Syntax□features consist of Part-of-Speech (POS) tags, as well as dependency parsing trees. We first parsed the input text using the Charniak's ME reranking parser[13] with its improved, self-trained biomedical parsing model[14]. The extracted POS features were then used in an analogous way to how we created the hybrid n-grams: words are substituted by their POS-tags. Then, the constituent parse trees were converted into Stanford dependencies[15] and features, similar to those we used in the non-temporal task[5], were extracted from the dependency paths. The improvement that syntax features brought over only lexical features indicates a correlation between syntax and temporal relations. Indeed, the word distance between the two extents following a parse tree path is more indicative for relationships than following the linear sentence structure, and therefore reduces noise in the feature set[5].

Semantic features:□The limited quantity of labeled data leads to a sparseness of features, which we attempt to reduce through smoothing by using manually created lexicons and knowledge bases. We incorporated two types of manually created lexicons. The first is a vocabulary of approximately 600 words/phrases related to temporal relations. We divided them into subcategories that express the concepts of *before*, *after*, *causing*, *caused by*, *during*, *starting*, *continuing*, *ending*, *suddenly*, and *currently*. If a word or a phrase is in the lexicon, it is substituted with its subcategory label (e.g., *causing*). We also incorporated UMLS mappings through MetaMap[16, 6] to represent words by their semantic categories, and used Metathesaurus label pairs associated with each word pair between the two extents, i.e., one label from each extent.□

Greedy **structural features**: up to now, relations are classified independently. We also adapted the method proposed by Roberts[17] to relieve this assumption. That is, when deciding a relation's category, the results/labels of the nested relations are also used as features. In this way, the temporal labels of closer extent pairs are utilized to guide the decisions made on the pairs that are farther apart; the latter are often more difficult to handle. In principle, this approach can be regarded as a greedy way to conduct structured learning to satisfy the constraints in the output space (here temporal labels).

Sectime Specialist features

The features for the Sectime specialist are described in full in the main article. For consistency, the full text is repeated here.

The annotation standard for the i2b2 task specifies that each event in a document must be connected to either the admission or discharge date, via a 'Sectime' link. An event in the patient history section links to the admission date, while an event in the hospital course section links to the discharge date. Because these Sectime TLINKs are constrained by very specific rules, we target them using a special-purpose classifier.

We trained a multi-class SVM to categorize each event into one of seven classes: BeforeAdmission, OverlapsAdmission, AfterAdmission, BeforeDischarge, OverlapsDischarge, AfterDischarge, Empty. We selected an SVM classifier over Maximum Entropy, because it achieved higher accuracies in cross-validation, and because precision is high enough that we did not feel we would benefit from inducing probabilities over classes. Our SVM is an in-house implementation, similar to LibLinear[18]. Cross-validated in isolation, this Specialist achieved 89.6% accuracy by constructing six binary feature templates from four binary template atoms:

- Bias: A feature that is always on
- Loc: Indicates whether the event appears in the patient history section, or the hospital course section. These two sections are separated by a 'hospital course' line, defined heuristically as the shortest line that matches the case-insensitive regular expression "hospital[ ]+course".

- LineNo: Indicates the line number of the event, divided into the following bins: 1, 2, 3, 4, 5, 6+.
- Str: Indicates the complete lowercased string corresponding to this event as it appears in the text.

The atoms were composed into the following six templates, where • indicates composition: Bias • Y, Loc • Y, LineNo • Y, Str • Y, Loc • LineNo • Y, Loc • Str • Y. Note that every feature is concatenated with the class Y to enable multi-class classification.

Non-local Event Overlap Specialist features

We train a binary maximum entropy classifier to assign each event pair to either an OVERLAP or OTHER class. Our features (Table A2) summarize the two events involved in terms of their local contexts and the words and extents that come between them. Any integer-valued features are binned into the following bins: 0, 1, 2, 3, 4, 5, ≤ 10, ≤ 25, >25 to create binary features. Features that track words have two versions: one that tracks the lowercased word, and another that tracks the word shape. Word shape maps a word to an alternate character representation, with all capital letters mapping to 'A', all lower case letters mapping to 'a' and all digits mapping to '0'; after the first three characters of a word, consecutive characters of the same shape are collapsed to a single shape. As an example, the word shape of 'pCO2' is 'aAA0'.

Table A2: Features for characterizing temporal *Non-local Event Overlap* relationships

Base features
- Types, attributes and binned lengths for the From and To extents (E1&E2) in a relation.
- Indicators on extent types and attributes for extents between E1&E2.
- Counts of extents between E1&E2.
- Words between E1&E2.
- The document section (patient history or hospital course) of E1&E2.
- Local context (words within a 2-word window) around E1&E2.
Features for cases where To and From extent strings match
- The entire event string, capitalization profile of the entire string.□
- Prefixes and suffixes of the event string (up to character length 5).□
- All word n-grams (up to n-gram length 5) of the event string.
Features for cases where To and From extent strings do not match
- The cross product of combining all word n-grams (up to 5-grams) in the E1 string with all word n-grams (up to 5) in the E2 string.□
- Word edit distance of E1&E2 strings.□
Concatenated to all the above features (each of the above has 3 augmented versions):
- Distance between E1&E2 in sentences.□
- Are E1&E2 an exact string match.□
- Extent types for E1&E2.□