Event ordering

1 Finding best event ordering

Process descriptions contain a series of events tied together by several relations. For instance, there could be a temporal ordering of events where one event can happen only after the other; or events can be overlapping at some point in time. In other cases, several events could have a super-event. In this section, we describe our models for used for predicting event ordering. The event triggers predicted earlier are ideally to be used in this stage. But, to start with, we will use the gold standard triggers for predicting the event ordering.

We use concepts based on structured probability models (graphical models) for the task at hand. Given a paragraph of text and the event triggers present in the paragraph, the goal is to generate the event ordering and relations between each pair of events into different categories - cotemporal, next event, super event etc. or NONE and come up with a global structure of events occuring in the paragraph. The problem is modeled as an inference problem in a Markov Network. The nodes are the triggers and energy terms are used to indicate interactions between triggers. We model local interactions between every pair of event triggers separately from those between groups of event triggers. For the local pair wise classification, we use a MaxEnt model that use lexical and structural features. However, there may be different global constraints that need to be satisfied, for instance, each event may have only one super event and next event. These kind of global constraints require a broader document context and it helps to assimilate the local decisions.

Hence, we have an undirected graphical model with a set of vertices $V = X \cup Y$. X is the set of observed nodes and x_i denotes the i^{th} trigger in the paragraph. Y is the set of unobserved nodes corresponding to the labeling for each pair of event trigger and y_{ij} is the labeling of the pair x_i and x_j. The number of triggers in the paragraph is denoted by n. We have two types of potentials:

1. *Event-pair Potential* associates two event triggers in a paragraph based on their local context.
2. *Global Consistency Potentials* are used to ensure global consistency of assignments and are defined over entire sets of variables in the paragraph.

The resulting MAP problem is:

$$MAP(\theta) = \sum_{f \in F} \theta_f(r_f) \quad (1)$$

where θ_f are the potential functions and $\{r_f | f \subseteq \{1, 2, ..., n\}, f \in F\}$ is the set of their variables.

1.1 Modeling local dependencies

Let E be the set of all n trigger words $\{e_1, e_2, ...e_n\}$ in the paragraph. The local event-event relation classifier algorithm generates a mapping R of all pairs of event triggers in E to exactly one event relation type or NONE, R : $E \times E \rightarrow$ { *NextEvent, SuperEvent, CotemporalEvent, NONE*}. Formally, our goal is to maximize the likelihood of the labeling R $P(R|E)$, given the set of triggers.

We trained a MaxEnt model on the annotated samples using several lexical, and structure based features.

$P(R|E) = \prod_{e_i \in E, e_j \in E} P(l_{ij}|x)$,
where l_{ij} is the labeling of the relation between e_i and e_j according to the mapping R.

The lexical features we use are the lemma and POS tags of the triggers and the words immediately before, after and between the triggers, the modal verbs to the left and right of event mentions, the temporal connectives between the event mentions. Syntactic features include which event appears first in the text, the number of words between the triggers, number of sentences between the triggers. In addition, to capture semantics, we also use the cluster IDs to which the event triggers belong to.

The potential functions of these components are given by the likelihoods of the corresponding labeling according to the MaxEnt model.

1.2 Modeling global dependencies

The main function of the global constraints is to ensure more probable assignments of all the event-event relations in addition to resolving potentially

inconsistent decisions generated by the local models. The global constraints help satisfy consistent assignments by encouraging the local assignments to agree with paragraph level properties that are global. We have two kinds of global potentials. The ones for hard constraints that are defined as:

$$\theta_f(y_{f-p}) = \begin{cases} 0 & if property holds \\ -\infty & otherwise \end{cases} \quad (2)$$

and the ones for soft constraints defined as:

$$\theta_f(y_{f-p}) = \begin{cases} \alpha_p & if property holds \\ 0 & otherwise \end{cases} \quad (3)$$

where $f - p$ is the index set of variables over which the potential is defined.

We use four global consistency potentials:

1. *Next event potential* is a soft constraint and is applied for each event. It is used to penalize if an event has more than one *NextEvent.*

2. *Super event potential* is a soft constraint and is applied for each event. It is used to penalize if event has more than one *SuperEvent.*

3. *Connected component potential* is applied for the full network to ensure that all events in the paragraph are form a single connected component of undirected edges. This is a hard constraint.

4. *Consistency potential* is applied for each pair of events and is used to ensure that there are no inconsistent assignments. For instance, two events cannot be both *NextEvent* of each other.

Dual Decomposition The global dependencies are very important as they ensure consistent and more likely global assignments. But, at the same time, enforcing these dependencies complicates the inference by a great extend as now we have to consider all possible assignments of all variables in the factors generated by the dependencies. Hence, we propose an approach based on dual-decomposition, which is an inference technique that helps to compute a tighet upper bound on the original MAP objective efficiently which preserving all the original dependencies. As a first step, we modify the MAP equation to include the local potentials for the event pairs.

$$MAP(\theta) = \max_y \sum_{j \in J} \theta_j(r_j) + \sum_{f \in F} \theta_f(r_f) \quad (4)$$

where the set of unobserved variables are denoted by J and each of the variables is denoted by r_j. The dual problem is now defined as:

$$\min_\delta L(\delta), L(\delta) = \sum_{j \in J} \max_{r_j} [\theta_j(r_j) + \sum_{f: j \in f} \delta_{fj}(r_j)] + \sum_{f \in F} \max_{r_f} [\theta_f(r_f) - \sum_{j \in f} \delta_{fj}(r_j)] \quad (5)$$

where for every $f \in F$ and $j \in f, \delta_{fj}$ is a vector of Lagrange multipliers with an entry for each possible assignment of r_j. Our goal is to find the tightest upper bound for the dual objective $L(\delta)$. This can be done efficiently by using subgradient descent algorithm.

Inference algorithm The dual decomposition algorithm and arg max computation algorithm we use are similar to Reichart and Barzilay.

Maximizing Individual Potentials For the local potentials rl_j^k, the maximizing assignments are found by decoding the likelihoods of the assignments and picking the most likely assignment from the sorted list of likelihoods from the MaxEnt model.

The global potentials are harder to compute. First step is to compute the *minimum-message assignment (MMA)* which minimizes the message sum. If this assignment is consistent with the potential property, then this is the best assignment. If not, we compute the *property-respecting assignment (PRA)*, which gives the lowest message sum under which the potential property holds. The best of MMA and PRA is selected by our algorithm.

MMA is the minimum message assignment of each unobserved variable in isolation. For PRA, we have to evalute over a very large number of assignments. We begin with the MMA and try to generate all possible combinations that satisfy a specific potential property.

- *Next event potential* For each event, we need to restrict the number of next events to at most 1. So, if there are more than one next-events for an event according to the local models, then we have to ensure that the constraint is satisfied by finding the best next-event to keep. This can be done by generating all options to keep exactly one of the

events tagged as next events while assigning the next-best option for the other events. This is followed by picking the best event to keep as the next-event.

- *Super event potential* This constraint can be done in a similar way to 'Next Event Potential'.

- *Connected component potential* This can be modeled by Kruskal's algorithm. The weight of an edge between two nodes is the score for the best non-NONE labelingbetween the two events. The algorithm could continue till there are no more positive edges to be added or the graph is not connected yet.

- *Consistency potential* This constraint can be satisfied by evaluating the relation between each pair of events and making sure that the assignments are consistent. For instance, two events can both not be next events of each other. To resolve inconsistencies, we pick the highest scoring consistent assignment for the pair by taking the second-best option for either. Since this would result in a change in labeling of one of the relations, we have to ensure that the new scheme is also consistent. This may result in cascading changes.