Event Extraction for Semantic Understanding

Aju Thalappillil Scaria, Rishita Anubhai, Rose Marie Philip
{ajuts, rishita, rosep}@stanford.edu

Project Goals

In the literature review, we focused on research publications in the areas of event and entity extraction with semantic role labeling and temporal ordering of events. Even though there has been a lot of work on one (or some) of these areas in isolation, there is no approach that seamlessly integrates all these components together. Through this project, we envision to build a system that takes a paragraph of text as input and does the following tasks:

1. Identify the events by locating the trigger words.
2. For each event, identify its arguments (only entities).
3. For each argument that is associated with a specific event, label the association with a semantic role, like Agent, Destination, Theme, Location etc.

Previous Work

Most of the work in the area of event and entity extraction can be categorized by different criteria:

- **Coverage and domain.** Most of the previous work dealt only with a subset of the tasks listed as the project goals and dealt with very specific domains which would not generalize.
- **Parsing scheme: Constituency vs Dependency parse.** Some of the previous work relied on the constituency parse structure of sentences, while others used just the dependency parse structure.
- **Modeling: Graph vs Tree.** Some of the earlier work is based on graphs and deals with edge prediction, while others use tree structure with classification of nodes.

Current Approach

In this project, we combine the learnings from the different methodologies to build a model that can be used for event and entity extraction with classes that are not specific to any domain. Even though our dataset is based on paragraphs from a biology textbook, we believe our models would generalize well to deal with more general content as our features and event/entity classes are not tied to the biological domain in any way. The key modeling decisions are as follows:

- We model events and entities as nodes in the constituency tree. Each sentence is assumed to be independent of each other as far as entity-event relationships are concerned. Events are denoted by their trigger word and are hence pre-terminals in the parse tree. Entities are denoted by a parse tree-node covering a sub-tree that spans over the whole text of the entity. In some cases when there is no single node that covers the entire entity (mostly because of parser errors, for e.g., PP attachment), we use an approximation by repeatedly removing tokens from the end or beginning of the span of text to identify a node that covers it. We manually verified that this heuristic works well in practice and results in entities that convey almost the full meaning of original span, and are well-formed.
- We also use features based on the dependency parse in conjunction with the constituency parse, since it contains information about the dependencies between tokens. This is done by identifying the position of the headword of an entity or event from the dependency tree and analyzing the relations.
- We handle task 1 as an independent classification task, while tasks 2 and 3 are designed as a joint classification task. For all classification tasks, we use maximum entropy model based on an implementation of L-BFGS for Quasi Newton unconstrained minimization.

Dataset and Evaluation

In this project, the dataset was prepared by annotating 125 paragraphs from different chapters of the text book *Biology (Eighth Edition)* by *Neil A. Campbell* and *Jane B. Reece*. Each paragraph is a text file and has an associated annotation file that indicates the different events and entities (by their character offsets in the original paragraph) and the event-entity and event-event relationships. The annotations were done by experts in the field (employees of the company Vulcan). Since there is not much data at our disposal, we split the data by a proportion of 70-30% for training and testing. We randomly permute the order of files to avoid similarities in adjacent files and then use 10 fold cross validation on the training set. For event prediction, we use F1 score based on the trigger predictions made. In entity prediction, the F1 score is based on whether an entity was predicted correctly along with its association with the corresponding event.

	Precision	Recall	F1
Baseline	0.47	0.72	0.57
MaxEnt-Train	0.86	0.71	0.77
MaxEnt-CV	0.71	0.67	0.69

Table 1: Event trigger prediction

	Precision	Recall	F1
Baseline	0.52	0.70	0.59
MaxEnt-Train	0.81	0.66	0.72
MaxEnt-CV	0.69	0.60	0.64

Table 2: Entity prediction for event triggers

Progress

In our project, we use Stanford Core NLP tools. We use the annotation pipeline available in the toolkit including tokenization, lemmatization, dependency and constituency parsers, and POS taggers. The events, entities and their relationships are represented as annotations on the already existing sentence annotations, by implementing the *CoreAnnotation* interface. This helps us to integrate our codebase with the existing features of the CoreNLP toolkit. We describe the progress made on the different tasks in this section.

1. **Event prediction.** As we mentioned earlier, events are represented as pre-terminal nodes in the parse tree of a sentence. As a first step to the task, we built a baseline model that predicted every pre-terminal node whose part-of-speech tag started with 'VB' to be an event trigger. This model performed quite well giving an F1 score of 0.565, considering that it was a very naive approach. As the next step, we designed a MaxEnt model that trained on the annotated samples using several lexical and path features. The features we currently have include part-of-speech tag of the word, its lemma, the part-of-speech tag of its parent, the actual word itself and the path from root to the node. The results we have are in Table 1. On doing error analysis, we found that our classifier fails to identify nominalized verb forms as event triggers. Even though we tried using a feature to indicate nominalization by looking up in a dictionary of nominalized verb forms, the classification accuracy did not improve, probably because they are common even in words that are not event triggers.

2. **Entity prediction and Semantic role labeling.** An entity is represented as a node in the parse tree spanning over the full text of the entity along the leaves of the tree. The fact that there is more than one event in almost all sentences makes our task of event-entity association harder. This is because, instead of just predicting a node in the parse tree as an entity, we have to predict if a node is associated with a specific trigger from task 1. To overcome this, we assign a probability for each node in the parse tree to be an entity associated with each event. Since this model was developed in parallel to the one in task 1, we are currently using the gold standard trigger words to denote events. Once we attain reasonable performance levels, we will use the predictions from step 1 to replace the gold standard. In addition, currently the model only tags if a node in the parse tree is associated with a specific event trigger or not. Since we are using a MaxEnt model, extending this to predict semantic role labeling would make it from a 2 class classification (Argument and None) to a multi-class classification (where the classes are the semantic roles like Agent, Theme, Destination, Origin, Result etc. and None).

   As a first step, we built a baseline model that predicts a node in the parse tree as an argument to an event trigger, if it is of part-of-speech tag 'NP' and if the head word of the node in the parse tree is a child of the event trigger in the dependency tree of the sentence. We used Collins head finder algorithm to identify the headword of a parse tree node. The baseline model intuitively captures the relation between event triggers and its arguments as is evident from the F1 score of 0.593 achieved using a relatively simple approach. We then implemented a MaxEnt based model using more features between the event triggers and the candidate nodes. The features we use include POS tag of node + POS tag of event trigger, headword of node + POS tag of event trigger, path from the node to the event trigger and an indicator feature denoting whether the headword of the node is a child of the trigger in the dependency tree. The results are presented in Table 2 .

   Dynamic Program for non-overlapping constraint. Since we predict a node in the parse tree as an entity or not, there are instances when predicted entities overlap. For instance, a sub-tree of a tree node already marked as an entity may also be tagged as an entity. To avoid this, we devised a bottom-up dynamic program that tags a node as entity or not, looking at the probability of the node and its immediate children being entities. There are two scenarios. If we tag the node as entity, none of its children can be entities. If we do not tag the node as an entity, then the children can retain their class (Entity or None). Another addition we did to the dynamic program was that an entity node cannot subsume a node in the parse tree that is an event trigger. The dynamic programming approach gave us a boost of 0.04 (0.60 to 0.64) in F1 score.

Next Steps

1. Since most of the errors made by event prediction model is related to verb nominalizations, extend the model to capture them as well.
2. Extend the entity prediction task for semantic role labeling. This would be an extension from binary classification to multiclass classification using the MaxEnt model. Depending on the performance of this model, we would also look at re-ranking models similar to Toutanova et. al.
3. We feel that knowing entities might help in improving our event trigger prediction. For this, we need features in task 1 that use information about entities in the sentence. We are thinking of an iterative solution similar to Expectation Maximization algorithm that would repeat tasks 1 and 2 to converge to the best event-entity extractions.