KU Leuven - NLP 2021-2022
21 December 2021

Exercise Session 6
Temporal Analysis and Text Spans

For questions contact:
Toledo Forum for Questions or
nlp@ls.kuleuven.be

1 Temporal Annotations

Allen's interval algebra consists of 13 temporal relations between intervals. In the Timebank Dense corpus (Cassidy et al., 2014), the text is annotated with 5 of these relations: *before*, *after*, *during*, *includes*, *simultaneous*. In this exercises, these are collapsed to three basic relations: *before*, *includes*, and *simultaneous*. These are enough, since after is the inverse of before and includes that of during.

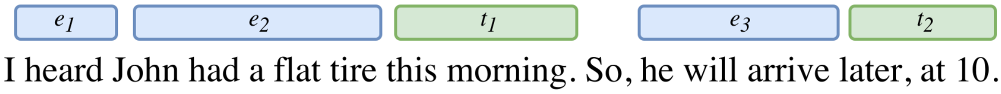


Figure 1: Example sentence annotated with events (blue) and temporal expressions (green).

1.A

Construct the temporal graph for the example sentence in Figure 1, annotated with events and temporal expressions, i.e. connect the colored boxes with labeled arrows of types: *before*, *includes*, *simultaneous*.

1.B

In Figure 2, two annotators annotated the same sentence with temporal relations. Calculate the inter-annotator agreement for between A_1 and A_2, using F1-measure, and check if the two annotations would result in different time-lines. Why do you think this is a problem?

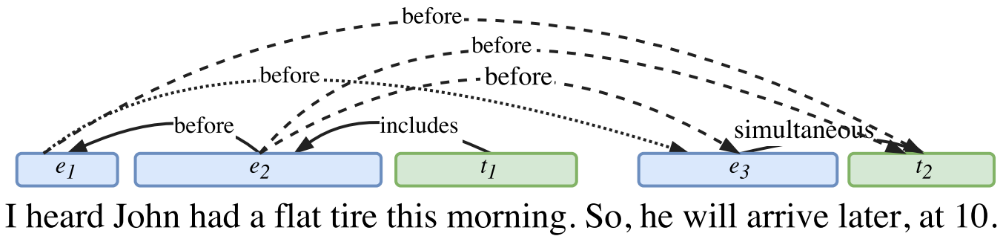


Figure 2: Relation annotation for two annotators A_1 (solid + dotted lines) and annotator A_2 (solid + dashed lines).

1.C

What rules can you come up with that infer the relations of A_2 from the relations of A_1?

2 Temporal Inference

In this section, we assume we have already trained a model that assigns probabilities $P(r|c_{ij}, s)$ to relation types $r \in \{before, includes, simultaneous, none\}$, for each candidate entity pair $c_{ij} \in E \cup T \times E \cup T$ in the sentence s, with E the events and T the temporal expression in the sentence. It uses a simple greedy inference to infer the predicted relations. The predicted relation $\hat{r}_{ij}$ for a candidate c_{ij} is the one that has the highest probability according to the model:

$$\hat{r}_{ij} = \arg\max_{r} P(r|c_{ij}, s)$$

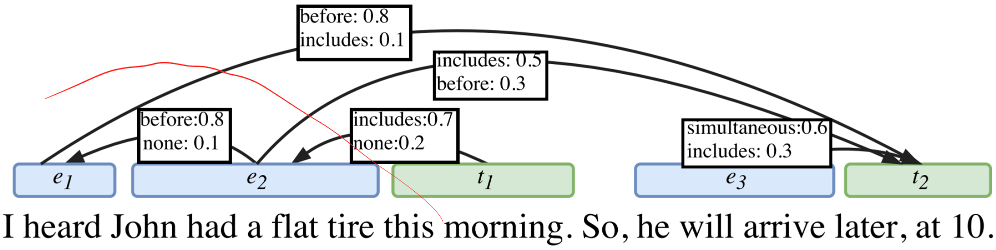


Figure 3: Example of top-2 highest probabilities given by the model P for the example sentence. Arrows for pairs for which the highest probability is given to the none-relation are not drawn.

2.A

What is the problem for the model output in Figure 3 if we use greedy inference?

2.B

We can improve the inference by informing it with the temporal rules we made up in Question 1.C, using integer linear programming (ILP). To use ILP for inference we need to construct a set of decision variables $r_{ij} \in D$ such that r_{ij} corresponds to assigning relation r to candidate pair c_{ij}. Additionally, we need to define an objective that, when maximized, gives us the desired values for D.
As an example, greedy inference can be formulated as ILP. The objective would be:

$$Objective = \max \sum_{c_{ij}}^{C} \sum_{r}^{R} r_{ij} \cdot P(r|c_{ij}, s)$$

However, we are not finished yet, as now it can be that $before_{ij} = 1$, $includes_{ij} = 1$, $simultaneous_{ij} = 1$, and $none_{ij} = 1$ at the same time. As that would maximize the objective.
To prevent this we need to add the following constraints to the integer linear program:

$$\forall_{c_{ij}} : \sum_{r}^{R} r_{ij} = 1$$

This makes sure only one relation is predicted per candidate.
Can you formulate the rules from Question 1.C as additional ILP constraints such that the model output in Figure 3 gives better predictions?

3 Using SpanBERT

In this question we will take a look at SpanBERT and start thinking about how to use the model. Figure 4 shows the architecture of the model and how it is trained.

$$\max \sum_{c_{ij}}^{C} \sum_{r}^{R} r_{ij} \, P(r|c_{ij}, s)$$

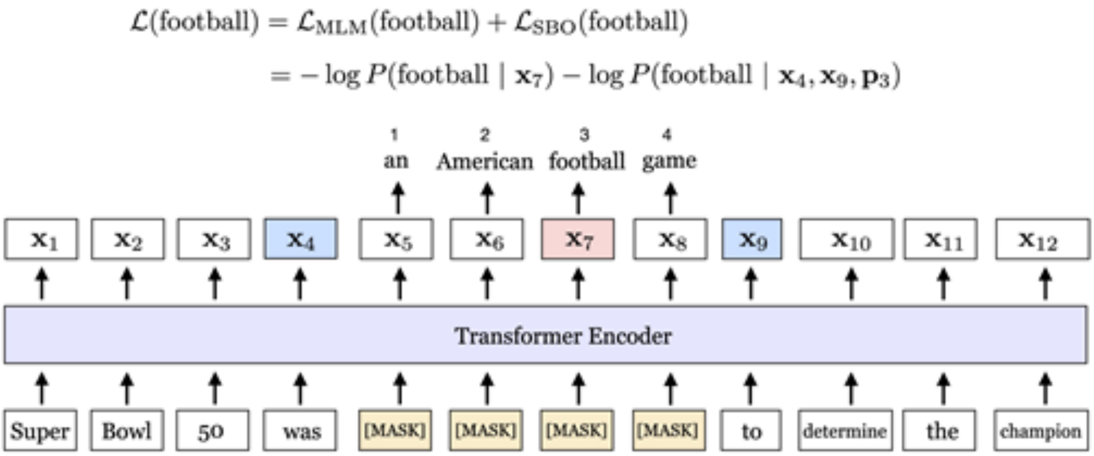


Figure 4: An illustration of SpanBERT training. The span an *American football game* is masked. The SBO uses the output representations of the boundary tokens, $\boldsymbol{x}_4$ and $\boldsymbol{x}_9$ (in blue), to predict each token in the masked span. The equation shows the MLM and SBO loss terms for predicting the token, *football* (in pink), which as marked by the position embedding $\boldsymbol{p}_3$, is the third token from x_4.

3.A: Training with masked spans

The model trains by masking around 15% of the words in the sentence, just as with BERT. The difference is that BERT masks random words, in SpanBERT we only want to mask spans.

Can you think of two methods on how to choose a span of words to be masked?

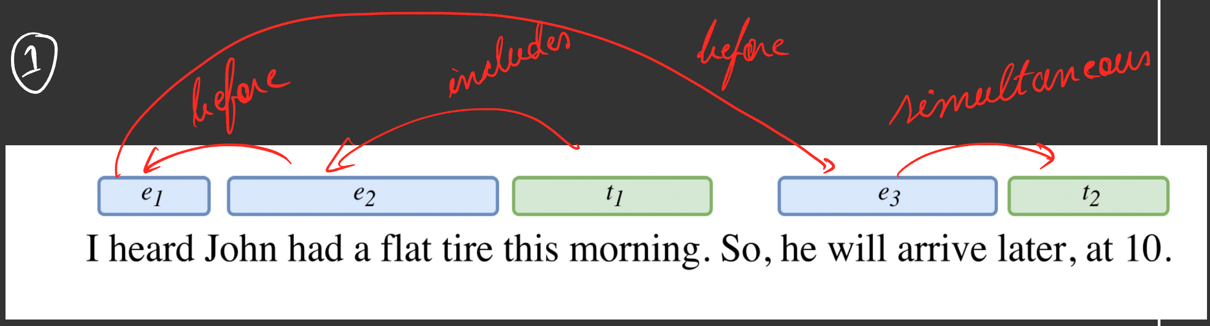
3.B: SpanBERT for SRL

We want to use SpanBERT for a different task, semantic role labeling. In this subquestion we will discuss if this would be a good approach to solving this task.
When the model receives a sentence and its predicate, the task of the model is to assign a label to each of the words in the sentence. The possible labels are:

```
[A0, A1, O]
```

1. Using your previous obtained knowledge about SRL, would it make sense to use SpanBERT? Do you think this would perform well?
2. Next we must decide how to train this model. We decided that it would be best to train it using a multitask objective.
   First, the model follows the span boundary objective (SBO) function. We randomly mask spans from the input, and the model must predict the words for the masked span.
   Next, after filling the mask with the predictions, we use cross entropy for making the categorical SRL predictions.
   Write down the complete multitask objective function.
3. We have seen such a method for predicting semantic roles and spans together before. How was this done in the lectures?

1



$$Precision = \frac{|A1 \cap A2|}{|A1|} = \frac{3}{4} = 0.75$$

$$Recall = \frac{|A1 \cap A2|}{|A2|} = \frac{3}{6} = 0.5$$

$$F1 = \frac{2 \times 0.75 \times 0.5}{0.75 + 0.5} = 0.6$$

$$before(i,k) \ \& \ before(j,i) \rightarrow before(j,k)$$

e_1 e_2