Document-level Event-based Extraction Using Generative Template-filling Transformers

Xinya Du, Alexander Rush, Claire Cardie

Overview

- This paper introduces a smart way based on seq2seq architecture to do event extraction at document-level.
- The proposed model is constructed completely based on the components of the pretrained BERT, NO extra parameters added.
- They propose to use an alternate metric for evaluation, which is slot-based, instead of mention-based as before.
- The proposed model significantly improves the performance with the proposed evaluation metric.

Overview

- Document-level event extraction:

Input document:

...
A bomb exploded in a Pilmai alley destroying some [**water pipes**].

According to unofficial reports, the bomb contained [**125 to 150 grams of TnT**] and was placed in the back of the [**Pilmai** [**telephone company building**]].

The explosion occurred at 2350 on 16 January, causing panic but no casualties.

The explosion caused damages to the [telephone company offices]. It also destroyed a [**public telephone booth**] and [water pipes].

Witnesses reported that the bomb was planted by [[**two men**] wearing sports clothes], who escaped into the night.
...
They were later identified as [[**Shining Path**] members].

Role	Role-filler Entities
Perpetrator Individual	**two men**, two men wearing sports clothes, Shining Path members
Perpetrator Organization	**Shining Path**
Physical Target	**water pipes**, water pipes
	Pilmai telephone company building, telephone company building, telephone company offices
	public telephone booth
Weapon	**125 to 150 grams of TnT**
Victim	-

Proposed Evaluation Metric

- "Document-level event-based template filling is ultimately an entity-based task"

Role	Role-filler Entities
Perpetrator Individual	**two men**, two men wearing sports clothes, Shining Path members
Perpetrator Organization	**Shining Path**
Physical Target	**water pipes,** water pipes
	Pilmai telephone company building, telephone company building, telephone company offices
	public telephone booth
Weapon	**125 to 150 grams of TnT**
Victim	-

Proposed Evaluation Metric

- Reference (gold) role-filler entities of one role in a document d:

$$R(d) = \{R_i : i = 1, 2, ..., |R(d)|\}$$

- Predicted role-filler entities:

$$S(d) = \{S_i : i = 1, 2, ..., |S(d)|\}$$

- Consider sets $R_m \subset R$, $S_m \subset S$ where $m = min(|R(d)|, |S(d)|)$
- G_m is the set of all possible one-to-one maps (of size-m) between subsets of R and S.
- First step: find best alignment between R and S:

$$g^* = \arg\max_{g \in G_m} \Phi(g) = \arg\max_{g \in G_m} \sum_{r \in R_m} \phi(r, g(r))$$

Where $\phi(r, s) = \begin{cases} 1, & \text{if } s \subseteq r \\ 0, & \text{otherwise} \end{cases}$

Proposed Evaluation Metric

- Second step: compute scores with the best alignment found:

$$\Phi(g^*) = \sum_{r \in R_m^*} \phi(r, g^*(r))$$

$$prec = \frac{\Phi(g^*)}{\sum_i \phi(S_i, S_i)}$$

$$recall = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)}$$

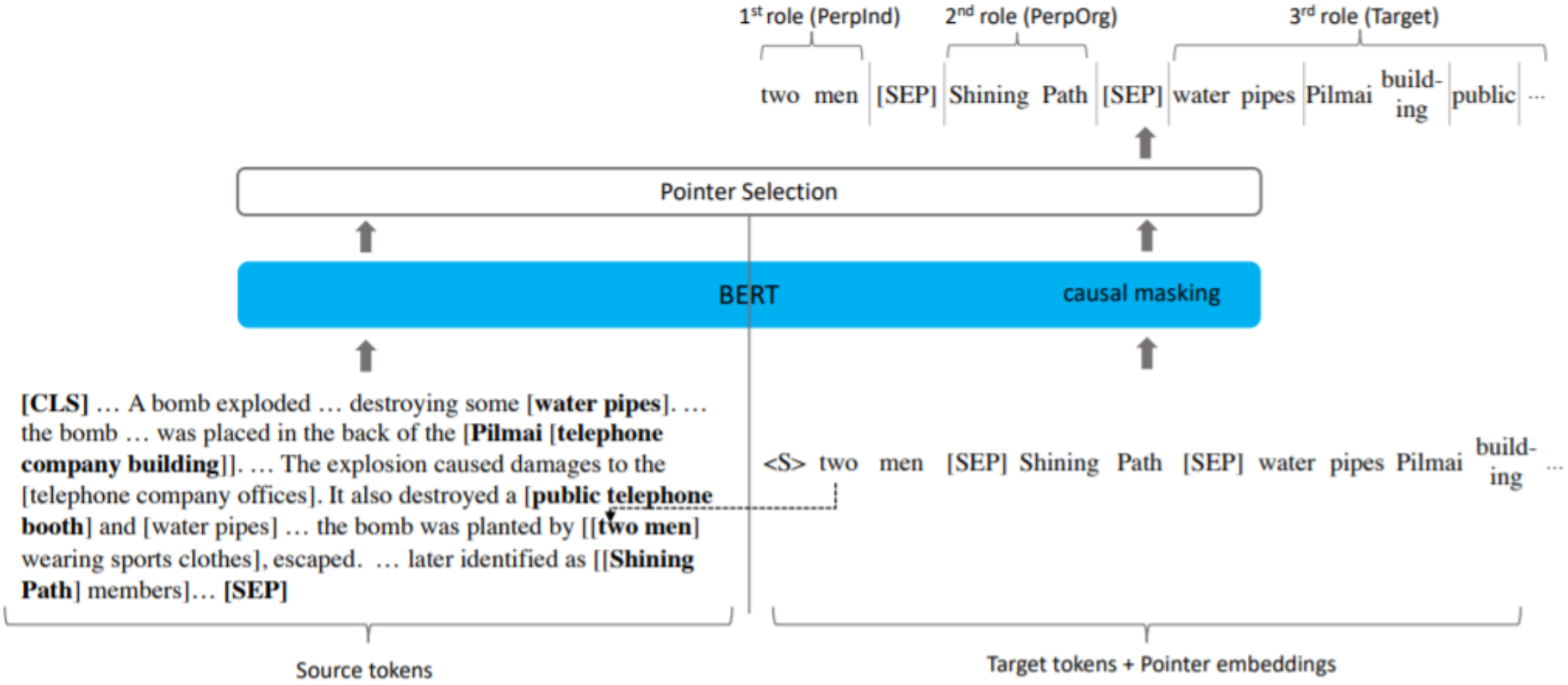
$$F = \frac{2 * prec * recall}{prec + recall}$$

- This new metric is based on CEAF, which is originally used for coreference resolution measure.

Model: Main Idea

- Use an encoder to encodes the context of the whole document (usually short in MUC dataset)
- Use a decoder to generate the answers for the slots of the template of the document.
- Modify pretrained BERT to make the encoder and the decoder work at the same place.

Model: Overall architecture



Model: Encoder & Decoder at the same place

- Source tokens: input document
- Target tokens: gold entities to fill in the given template, the entities are arranged in the predefined order which expresses which entities are used to fill which role:

$$\text{<S>}\ e^{(1)}_{1_b}, e^{(1)}_{1_e}, \ldots\ \text{[SEP]}$$
$$e^{(2)}_{1_b}, e^{(2)}_{1_e}, \ldots\ \text{[SEP]}$$
$$e^{(3)}_{1_b}, e^{(3)}_{1_e}, e^{(3)}_{2_b}, e^{(3)}_{2_e}, \ldots\ \text{[SEP]}$$
$$\ldots$$

Model: Autogressive masking

- They design a special mask for running the encoder and the decoder at the same place (i.e., BERT):

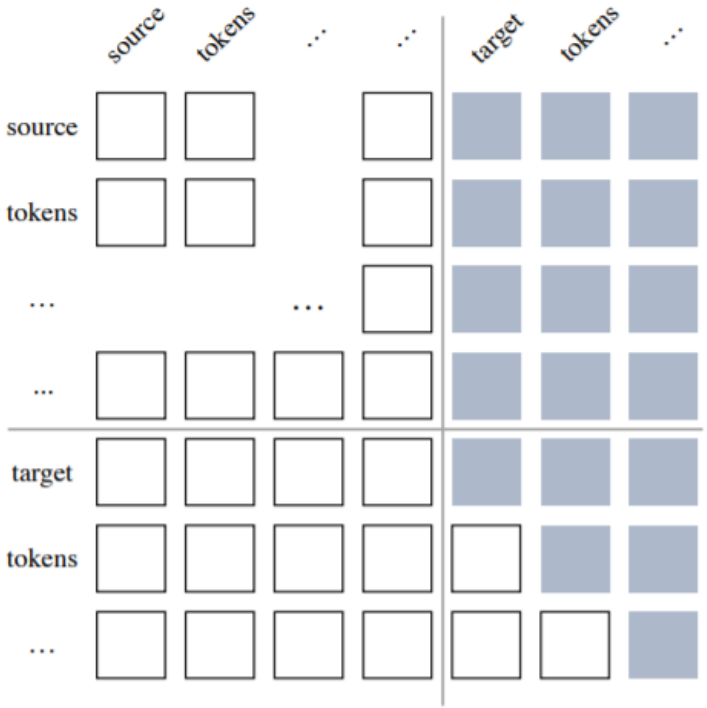
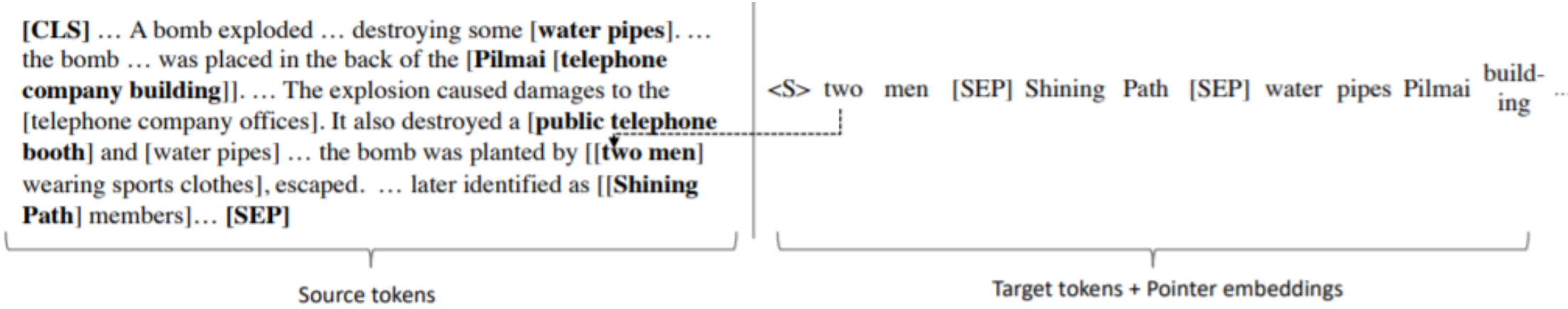


Figure 3: Partially causal masking strategy (M). (White cell: unmasked; Grey cell: masked).

Model: Pointer Embeddings

- To help the decoder be aware of the positions of the extracted entities in the source document, they use pointer embeddings which are positional embeddings of the extracted source tokens.



Model: Pointer Decoding

- At timestep t, compute the dot product of the current target token embedding with the source token embeddings:

$$z_0, z_1, ..., z_m = \hat{\mathbf{y}}_t \cdot \hat{\mathbf{x}}_0, \hat{\mathbf{y}}_t \cdot \hat{\mathbf{x}}_1, ..., \hat{\mathbf{y}}_t \cdot \hat{\mathbf{x}}_m$$

- Leverages the token classifier of BERT, which is already pretrained with th e Masked Language Modeling task to make prediction:

$$p_0, p_1, ..., p_m = \text{softmax}(z_0, z_1, ..., z_m)$$

Results

	PERPIND	PERPORG	TARGET	VICTIM	WEAPON
NST (Du and Cardie, 2020)	48.39 / 32.61 / 38.96	60.00 / 43.90 / 50.70	54.96 / 52.94 / **53.93**	62.50 / 63.16 / 62.83	61.67 / 61.67 / **61.67**
DYGIE++ (Wadden et al., 2019)	59.49 / 34.06 / 43.32	56.00 / 34.15 / 42.42	53.49 / 50.74 / 52.08	60.00 / 66.32 / 63.00	57.14 / 53.33 / 55.17
GTT	65.48 / 39.86 / **49.55**	66.04 / 42.68 / **51.85**	55.05 / 44.12 / 48.98	76.32 / 61.05 / **67.84**	61.82 / 56.67 / 59.13

Table 1: Per-role performance scored by CEAF-TF (reported as P/R/F1, highest F1 for each role are boldfaced).

Results

Models	P	R	F1
NST (Du and Cardie, 2020)	56.82	**48.92**	52.58
DyGIE++ (Wadden et al., 2019)	57.04	46.77	51.40
GTT	**64.19****	47.36	**54.50***

Table 2: Micro-average results measured by CEAF-TF (the highest number of each column is boldfaced). Stat. significance is indicated with $^{**}(p < 0.01)$, $^{*}(p < 0.1)$. All significance tests are computed using the paired bootstrap procedure (Berg-Kirkpatrick et al., 2012).