Span-based Joint Entity and Relation Extraction with Transformer Pre-training

Markus Eberts and **Adrian Ulges**[1]

这篇其实跟上一篇一样，也是为了解决NER的实体重叠问题，转而搞起了span-based。

感觉大体上没什么差异，只不过是模型结构与编码方法的变化。

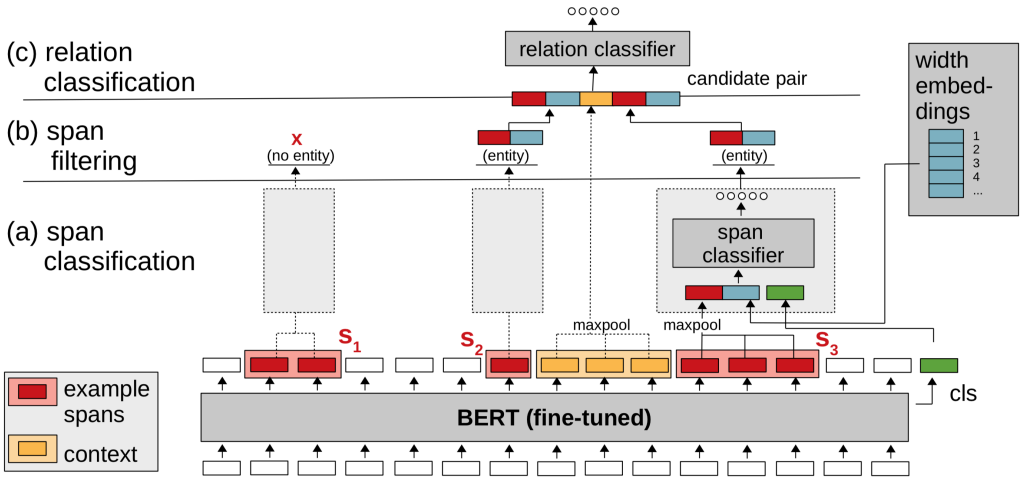

Figure 1. Our approach towards joint entity and relation extraction SpERT first passes a token sequence through BERT. Then, (a) all spans within the sentence are classified into entity types, as illustrated for three sample spans s_1, s_2, s_3 (red). (b) Spans classified as non-entites (here, s_1) are filtered. (c) All pairs of remaining entities (here, (s_2, s_3)) are combined with their context (the spa… the entities, yellow) and classified into relations.

（1）span classification

对于一个span要怎么分类呢?

首先经过BERT的encode，然后把这个span的embedding全部聚合起来，paper用的是max-pooling

然后concat上一个表示这个span长度的向量（类似于look-up table）

$$\mathbf{e}(s) := f(\mathbf{e}_i, \mathbf{e}_{i+1}, \ldots, \mathbf{e}_{i+k}) \circ \mathbf{w}_{k+1}.$$

这个就是这个span的representation了。

为了分类，就把这个span的representation concat上context的表示，这个用的是[cls]对应的输出，用它来代表整个sentence的信息
之后通过一个softmax来进行分类（这个entity是什么类型的entity，或者是NOne）

$$\mathbf{x}^s := \mathbf{e}(s) \circ \mathbf{c}$$

$$\hat{\mathbf{y}}^s = \text{softmax}\left(W^s \cdot \mathbf{x}^s + \mathbf{b}^s\right)$$

（2）Span FIltering
把none的span丢掉，然后其实也limit了span的长度最大是10.
paper说这样能够让时间复杂度是O(n)。
我想了一下，如果长度L，那么有L(L+1)/2 个可能，所以O(n^2)
如果变成10，那么O(10n) = O(n)
那这么说，前面那篇 span-level的实验的时候也limit了，那也是O(n)咯。

（3）relation classification
首先把两个span的representation e(s1) e(s2)拿过来
然后呢，paper提出，用这两个span中间的context（max-pooling之）比用整个句子context更好

$$\mathbf{x}_1^r := \mathbf{e}(s_1) \circ \mathbf{c}(s_1, s_2) \circ \mathbf{e}(s_2)$$
$$\mathbf{x}_2^r := \mathbf{e}(s_2) \circ \mathbf{c}(s_1, s_2) \circ \mathbf{e}(s_1).$$

Both $\mathbf{x}_1^r$ and $\mathbf{x}_2^r$ are passed through a single-layer classifier:

$$\hat{\mathbf{y}}_{1/2}^r := \sigma\left(W^r \cdot \mathbf{x}_{1/2}^r + \mathbf{b}^r\right)$$

实验再：

- CoNLL04
- SciERC
- ADE

我觉得比较没谱的是：为啥不跟上一篇那个再ACE2005比一下？
感觉两篇差不多啊，只不过这篇用了BERT。
其实本质上就是span + fine-tune BERT吧？