A Appendices

A.1 Hyper-parameter searching

We manually tune the hyper-parameters according to the performance of the model, i.e. the dev F1 scores. The hyper-parameters include the number of the stacked BiLSTM layers, the number of RoBERTa layers, and the subword pooling (use it or not). For the encoders or word embeddings, we used GloVe (Pennington et al., 2014), ELMo (Peters et al., 2018), SpanBERT (Joshi et al., 2020), XLNet (Yang et al., 2019), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019). After picking RoBERTa, we tried 1, 2, and 3 layers of stacked LSTM layers, and 1, 5, 10, 15, 20, 24 layers of RoBERTa. Among these trials, the model we adopt is with 2 layers of BiLSTM encoder and decoder, 24 layers of RoBERTa, and no subword pooling. Finally, we also benchmarked decoding strategy with beam search. With beam size 10, we gained 0.4 F1 over our greedy model but almost 70x slower. We leave *efficient* decoding strategies to future work.

The searching procedure and the intermediate results are shown in Table 3, Table 4, and Table 5.

	P	R	F1
Subword pooling	89.5	79.5	84.2
No pooling	89.6	84.5	87.0

Table 3: Comparison between using subword embeddings generated by RoBERTa directly, or pooling them into tokens representations, as evaluated on the NNE development set.

Number of layers	P	R	F1
1	90.6	82.6	86.4
2	89.6	84.5	87.0
3	87.8	83.1	85.4

Table 4: Performance of models with different numbers of layers in stacked BiLSTM encoder and decoder (on NNE development set).

A.2 Data

The nested named entity dataset is available online at https://github.com/nickyringland/nested_named_entities. Following Ringland et al. (2019), we use section 02 for development (1,989 sentences), sections 23 and 24 for testing (3,762 sentences), and the remaining sections for training (43,457 sentences).

layer	P	R	F1
1	86.4	73.9	79.7
5	89.0	80.2	84.4
10	90.9	82.3	86.4
15	91.3	83.2	87.1
20	91.0	81.0	85.7
24 (last)	90.6	82.6	86.4

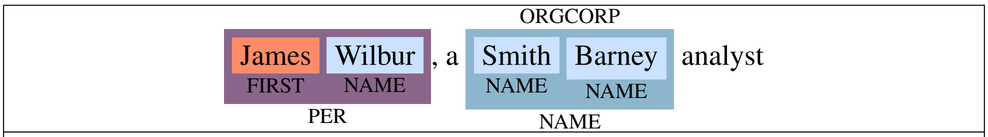
Table 5: Performance on the NNE development set using different layers of RoBERTa large as the input representation.

A.3 Example of linearization

Table 6 highlights another example of our linearization strategy. In our final strategy, ties are broken randomly when spans have multiple labels (such as “Smith Barney”) in the example. We did try sorting those spans by some deterministic method, such as label frequency (in the training corpus). We found that deterministically sorting these did not improve performance, sometimes even hurting.

A.4 Examples of errors

We present several examples of errors in Table 7. There are four major types. Two types are partially correct: (1) correct span boundary prediction but incorrect label; (2) incorrect span boundaries (still overlaps heavily with the correct span) but with the correct label. The other two types are (3) incorrect span and label, which combines both of the above errors, and (4) missing span entirely. Error types (2), (3), and (4) all affect recall (specifically span recall) and could provide further insight on how to improve our model’s recall. We did not find many instances where spurious spans are predicted.



0 CN PER 0 FIRST 1 NAME 4 CN NAME 4 CN ORGCORP 4 NAME 5 NAME

Table 6: Example of linearization of a structured output of nested named entities. Spans are in ascending order of starting index and ties are broken by span length.

Correct span, incorrect label	
predict	... to be the case in COUNTRY Sing apore , a country of about three million people with a rel atively high soft - dr ink cons umption rate – a key ind icator of C oke NAME ’s success in a market .
gold	... to be the case in CITYSTATE Sing apore , a country of about three million people with a rel atively high soft - dr ink cons umption rate – a key ind icator of C oke ORGCORP ’s success in a market .
Incorrect span, correct label	
predict	“ Nothing can be better than this , ” s ays Don S ider , owner of the West Pal m CITY Be ach T rop ics .
gold	“ Nothing can be better than this , ” s ays Don S ider , owner of the West Pal m Be ach CITY T rop ics .
Incorrect span and label	
predict	Pro ct er Gam ble Co . recent ly introdu ced ref ill able versions of four products including T ide and Mr . NAME Clean , in Canada , but doesn ’t plan to bring them to the U . S . .
gold	Pro ct er Gam ble Co . recent ly introdu ced ref ill able versions of four products including T ide and ANIMATE Mr . Clean , in Canada , but doesn ’t plan to bring them to the U . S . .
Missing entities	
predict	C ERT IFIC ATES OF DE POS IT : 8 . 09 % one month ; 8 . 04 % two months ; 8 . 03 % three months ; 7 . 96 % six months ; 7 . 92 % one year .
gold	C ERT IFIC ATES OF DE POS IT : 8 . 09 % CARDINAL one month ; PERCENT 8 . 04 CARDINAL % two months ; 8 . 03 UNIT % three months ; 7 . 96 % six months ; 7 . 92 % one year .

Table 7: Different types of error made by the CopyNext model in the NNER task.