

| No. | Model Name | Word Representation | Top Layer | Decoding Layer | F1 Score (\pm std) |
|-----|----------------------|---------------------|-----------|----------------|-----------------------|
| 1 | CNN-BLSTM-CRF | CNN-BLSTM | CRF | CRF | 90.92 \pm 0.08 |
| 2 | CNN-BLSTM-GSCRF | CNN-BLSTM | GSCRF | GSCRF | 90.96 \pm 0.12 |
| 3 | CNN-BLSTM-HSCRF | CNN-BLSTM | HSCRF | HSCRF | 91.10 \pm 0.12 |
| 4 | CNN-BLSTM-JNT(CRF) | CNN-BLSTM | CRF+HSCRF | CRF | 91.08 \pm 0.12 |
| 5 | CNN-BLSTM-JNT(HSCRF) | CNN-BLSTM | CRF+HSCRF | HSCRF | 91.20 \pm 0.10 |
| 6 | CNN-BLSTM-JNT(JNT) | CNN-BLSTM | CRF+HSCRF | CRF+HSCRF | 91.26 \pm 0.10 |
| 7 | LM-BLSTM-CRF | LM-BLSTM | CRF | CRF | 91.17 \pm 0.11 |
| 8 | LM-BLSTM-GSCRF | LM-BLSTM | GSCRF | GSCRF | 91.06 \pm 0.05 |
| 9 | LM-BLSTM-HSCRF | LM-BLSTM | HSCRF | HSCRF | 91.27 \pm 0.08 |
| 10 | LM-BLSTM-JNT(CRF) | LM-BLSTM | CRF+HSCRF | CRF | 91.24 \pm 0.07 |
| 11 | LM-BLSTM-JNT(HSCRF) | LM-BLSTM | CRF+HSCRF | HSCRF | 91.34 \pm 0.10 |
| 12 | LM-BLSTM-JNT(JNT) | LM-BLSTM | CRF+HSCRF | CRF+HSCRF | 91.38 \pm 0.10 |

Table 1: Model descriptions and their performance on CoNLL 2003 NER task.

| Component | Parameter | Value |
|--|---------------|---------------|
| word-level embedding [†] | dimension | 100 |
| character-level embedding [†] | dimension | 30 |
| character-level LSTM | depth | 1 |
| highway network | hidden size | 300 |
| word-level BLSTM | layer | 2 |
| word-level BLSTM | depth | 1 |
| word-level BLSTM | hidden size | 300 |
| CNN | depth | 1 |
| CNN | hidden size | 200 |
| CNN | window size | 3 |
| $\phi(\cdot)$ [†] | filter number | 30 |
| dropout [†] | dimension | 10 |
| dropout [†] | dropout rate | 0.5 |
| optimization [†] | learning rate | 0.01 |
| optimization [†] | batch size | 10 |
| optimization [†] | strategy | SGD |
| optimization [†] | gradient clip | 5.0 |
| optimization [†] | decay rate | $1/(1+0.05t)$ |

Table 2: Hyper-parameters of the models built in our experiments, where [†] indicates the ones when using LM-BLSTM for deriving word representations and [‡] indicates the ones when using CNN-BLSTM.

In the NER models listed in Table 3, [Zhuo et al. \(2016\)](#) employed some manual features and calculated segment scores by grConv for SCRF. [Lample et al. \(2016\)](#) and [Ma and Hovy \(2016\)](#) constructed character-level encodings using BLSTM and CNN respectively, and concatenated them with word embeddings. Then, the same BLSTM-CRF architecture was adopted in both models. [Rei \(2017\)](#) fed word embeddings into LSTM to obtain the word representations for CRF decoding and to predict the next word simultaneously. Similarly, [Liu et al. \(2018\)](#) input characters into LSTM to predict the next character and to get the character-level encoding for each word.

| Model | Test Set F1 Score | |
|--|-------------------|------------------------------------|
| | Type | Value (\pm std) |
| Zhuo et al. (2016) | reported | 88.12 |
| Lample et al. (2016) | reported | 90.94 |
| Ma and Hovy (2016) | reported | 91.21 |
| Rei (2017) | reported | 86.26 |
| Liu et al. (2018) | mean | 91.24 \pm 0.12 |
| | max | 91.35 |
| CNN-BLSTM-CRF | mean | 90.92 \pm 0.08 |
| | max | 91.04 |
| LM-BLSTM-CRF | mean | 91.17 \pm 0.11 |
| | max | 91.30 |
| CNN-BLSTM-JNT(JNT) | mean | 91.26 \pm 0.10 |
| | max | 91.41 |
| LM-BLSTM-JNT(JNT) | mean | 91.38 \pm 0.10 |
| | max | 91.53 |
| Luo et al. (2015)* | reported | 91.2 |
| Chiu and Nichols (2016)* | reported | 91.62 \pm 0.33 |
| Tran et al. (2017)* | reported | 91.66 |
| Peters et al. (2017)* | reported | 91.93 \pm 0.19 |
| Yang et al. (2017)* | reported | 91.26 |

Table 3: Comparison with existing work on CoNLL 2003 NER task. The models labelled with * utilized external knowledge beside CoNLL 2003 training set and pre-trained word embeddings.

Some of the models listed in Table 3 utilized external knowledge beside CoNLL 2003 training set and pre-trained word embeddings. [Luo et al. \(2015\)](#) proposed JERL model, which was trained on both NER and entity linking tasks simultaneously. [Chiu and Nichols \(2016\)](#) employed lexicon features from DBpedia ([Auer et al., 2007](#)). [Tran et al. \(2017\)](#) and [Peters et al. \(2017\)](#) utilized pre-trained language models from large corpus to model word representations. [Yang et al. \(2017\)](#) utilized transfer learning to obtain shared information from other tasks, such as chunking and POS tagging, for word representations.

From Table 3, we can see that our CNN-BLSTM-JNT and LM-BLSTM-JNT models with