Overview

- **Named Entity Recognition**: locate and classify text segments into pre-defined categories such as person, location etc.

- **Chinese NER**: character information and word information
  - **Character-based models**: take the character sequence as input, then label each character.
  - **Word-based models**: text are segmented as word sequence and label each word.
  - **Lattice models**: Character-based model with word lattice shortcut connection. Interaction of both word and character sequence.

Models

- **LSTM (coupled)**
  - Char-based and word based models have the same structure:
    $$i_j = \sigma \left( \tanh(\mathbf{W}_i x_{t-1} + \mathbf{b}_i) \right) + i_{j-1} + o_{j-1}$$
    $$c_j = (1 - i_j) \odot c_{j-1} + i_j \odot c_{j-1}$$
    $$h_j = o_j \odot \tanh(c_j)$$

- **Lattice LSTM**
  - Lattice path calculation:
    $$i_j = \sigma \left( \tanh(\mathbf{W}_i x_{t-1} + \mathbf{b}_i) \right)$$
    $$c_j = \sum_{k \in C(k \leq j \leq k_0)} \alpha_{kj} \odot c_{k_j} + \beta_{kj}$$
    $$h_j = o_j \odot \tanh(c_j)$$

- **Lattice & Character calculation**
  - use multiple normalized gates to control the contributions of different lattice paths.

Baselines: LSTM+CRF

- **Word baselines**: word-based LSTM+CRF models
  - +char LSTM: with extra char LSTM to represent word.
  - +char+bichar+LSTM: extra char+bi char LSTM to represent word.
  - +char+CNN: with extra char CNN to represent word.
  - +char+bichar+CNN: extra char+bichar CNN to represent word.

- **Character baselines**: character-based LSTM+CRF models
  - +softword: auto-segmentation results as neural features.
  - +bichar: with extra char bigram embeddings.
  - +bichar+softword: with both extra bigram and softword features.

Results

- **Dev Results**:
  - **Char-based NER**: char+bichar+softword gives the best result.
  - **Word-based NER**: word+char+bichar LSTM gives the best result.
  - **Lattice NER**: significantly improves the accuracy compared with both char-based and word-based baselines.
  - **Char vs Lattice**: char bigram information is useful in char-based baseline, while it does not improve the accuracy of lattice LSTM.

Experiments

- **Datasets**: four Chinese NER datasets
  - OntoNotes 4: news domain, with 4 entity types.
  - MSRA: news domain, with 3 entity types.
  - Weibo NER: social media NER corpus.
  - Resume NER: manual annotated, with 8 entity types.

- **Segmentation**:
  - Segmentor: SOTA word segmentor in Yang et al. ACL 2017
  - Lexicon/Word embeddings: auto-segmented Chinese Gigaword with the above segmentor and trained with word2vec.

Analysis

- **F1 with Sentence Length**:
  - **Char baseline**: is not sensitive with the sentence length.
  - **Word baseline**: works worse with the increase of sentence length, since the segmentor accuracy is worse in long sentences.
  - **Lattice LSTM**: In general, it gives better performance in all sentence length. It also suffers the accuracy deduction in long sentences, which can result from an exponentially increasing number of word combination in the lattice.