Semi-supervised Multitask Learning for Sequence Labeling

Marek Rei
University of Cambridge

Sequence Labeling

The task:

Given a sequence of tokens, predict a label for every token.

Named Entity Recognition:

```
PER _      _   _      _  ORG  ORG   _  TIME _
Jim bought 300 shares of Acme Corp. in 2006 .
```

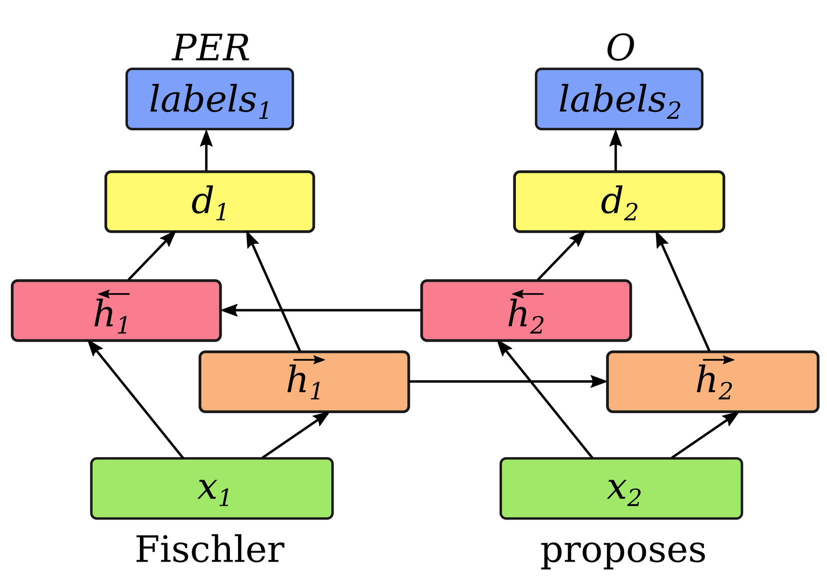
POS-tagging:

```
DT  NN    VBD      NNS    IN      DT  NN     .
The pound extended losses against the dollar .
```

Error Detection:

```
+ +    +  -       +   +      +   +    -      +
I like to playing the guitar and sing louder .
```

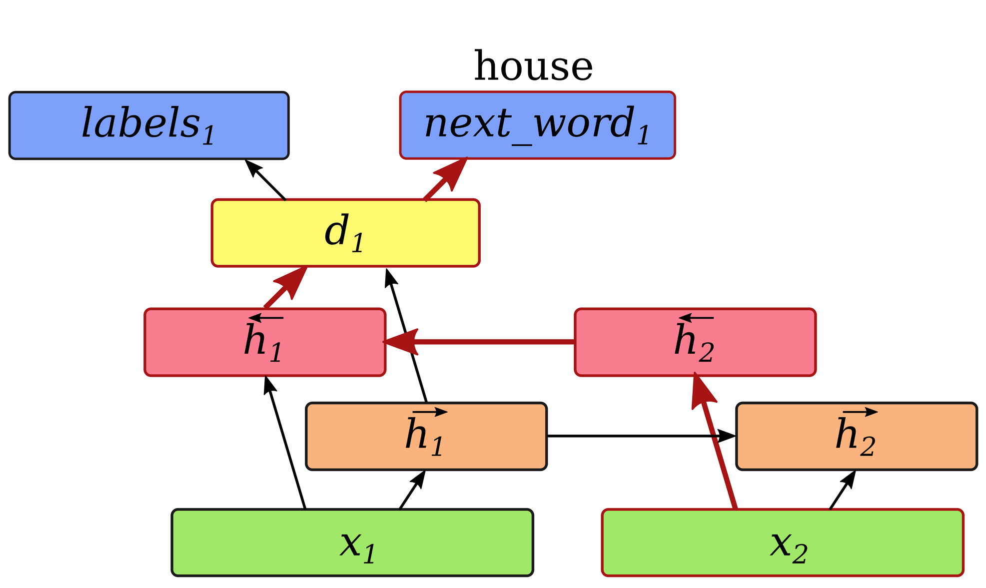
Neural Sequence Labeling



- **Sequence of tokens** mapped to word embeddings.
- **Bidirectional LSTM** builds context-dependent representations for each word.
- A small **feedforward layer** encourages generalisation.
- **Conditional Random Field** (CRF) at the top outputs the most optimal label sequence for the sentence.
- Using **character-based** dynamic embeddings (Rei et al., 2016) to capture morphological patterns and unseen words.

Multitask Learning

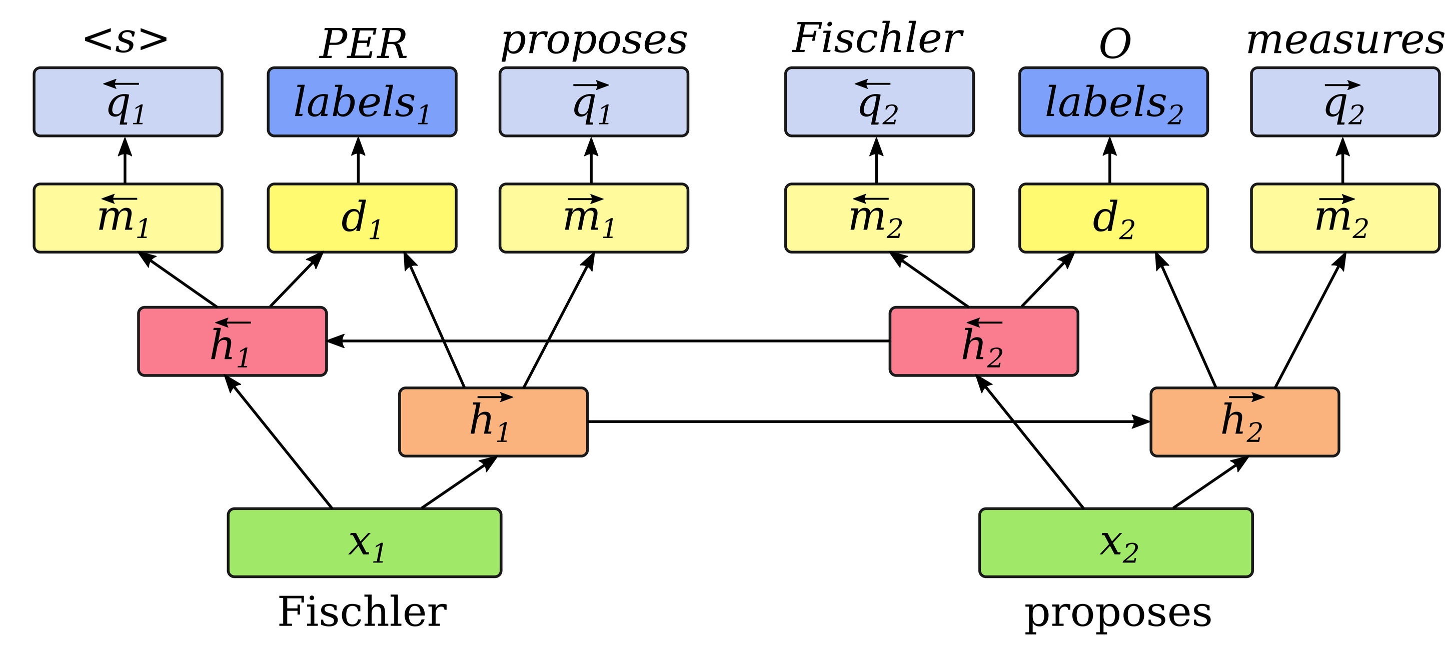
- Sequence labeling datasets can be **very sparse**: only 17% of tokens in CoNLL-03 are a named entity.
- We want an **additional objective** that makes full use of the data to learn features for semantic composition.
- **Language modeling** 1) requires no extra annotation, 2) has a large number of possible targets for each position.



- The network predicts the **next word** together with the main label.
- Cannot simply add it as an extra output layer – the next word is **already given as input** to the nextwork.

Language Modeling Objective

- The forward-moving LSTM predicts the **next word** in the sequence.
- The backwards-moving LSTM predicts the **previous word** in the sequence.
- Both LSTMs predict the **target label**.



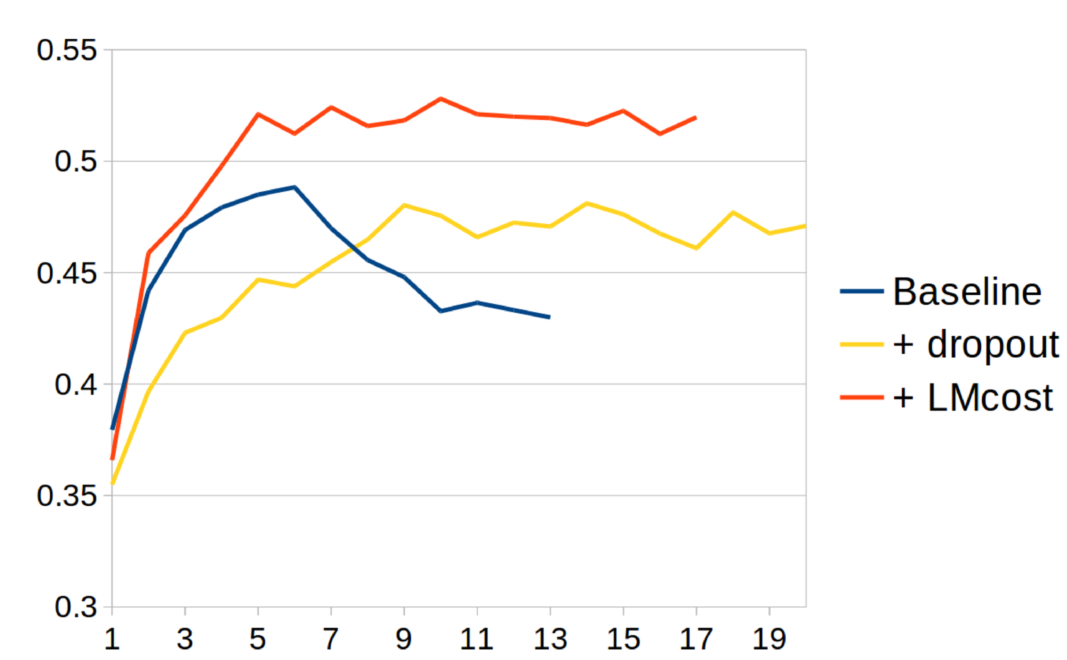
- The log-likelihood loss for both language models is added to the **training objective**:

$$\widetilde{E} = E + \gamma(\overrightarrow{E} + \overleftarrow{E})$$

$$\overrightarrow{E} = -\sum_{t=1}^{T-1} log(P(w_{t+1}|\overrightarrow{m_t})) \qquad \overleftarrow{E} = -\sum_{t=2}^{T} log(P(w_{t-1}|\overleftarrow{m_t}))$$

Analysis

- Visualising **convergence** on the FCE development set after each training epoch.
- LM objective improves performance at **all stages** of training.



- Additional **parameter matrices** are required for the two language models during training.
- However, the LM components are **not needed during testing**.
- The resulting model has the same structure and **the same number of parameters** as the baseline.

Conclusion

- Integrated a **language modeling objective** into a neural sequence labeling architecture.
- Requires **no additional data** and the trained model has no additional parameters.
- Provides **consistent improvements** on 10 different datasets.
- The **source code**: https://github.com/marekrei/sequence-labeler

Results

- Experiments on **10 different datasets** and 4 different tasks: error detection, named entity recognition, chunking, and POS tagging.

	FCE		CoNLL-14		CoNLL-03		CHEMDNER		CoNLL-00		PTB-POS		UD-ES		UD-FI	
	DEV	TEST	TEST1	TEST2	DEV	TEST	DEV	TEST	DEV	TEST	DEV	TEST	DEV	TEST	DEV	TEST
Baseline	48.78	44.56	15.80	23.62	90.85	85.63	83.63	84.51	92.92	92.67	97.23	97.24	96.38	95.99	95.02	94.80
+ dropout	48.68	42.65	14.71	21.91	91.14	86.00	84.78	85.67	93.40	93.15	97.36	97.30	96.51	96.16	95.88	95.60
+ LMcost	**53.17**	**48.48**	**17.86**	**25.88**	**91.48**	**86.26**	**85.45**	**86.27**	**94.22**	**93.88**	**97.48**	**97.43**	**96.62**	**96.21**	**96.14**	**95.88**