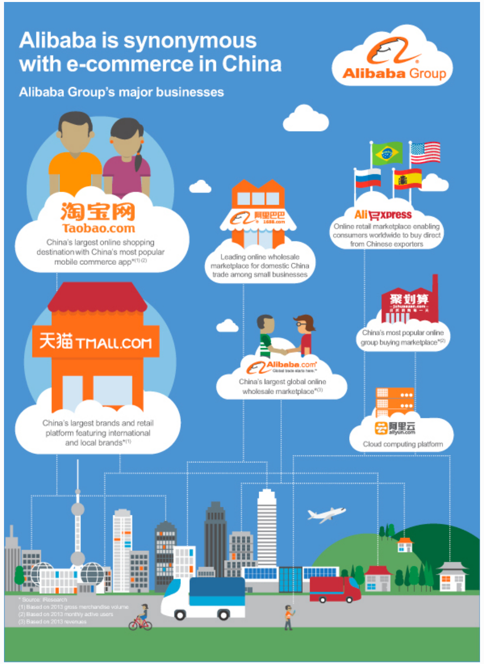
A Collection of Techniques for Improving Neural Entity Detection and Classification

Huasha Zhao, Yi Yang, Qiong Zhang, Luo Si
huasha.zhao@alibaba-inc.com
iDST, Alibaba Group
San Mateo, CA

Define Smarter Tomorrow.

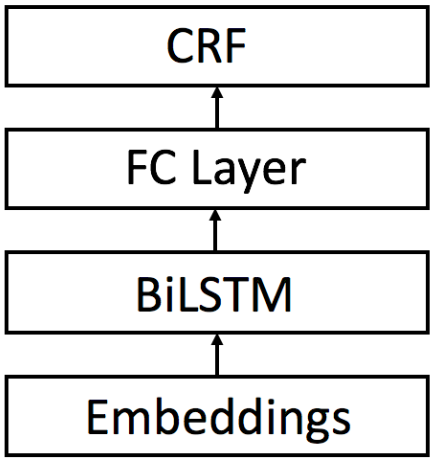
Agenda

- **Introduction**: Bidirectional LSTM-CRF
- **Features**: Multi-Input Model
- **Training**: Multi-Task Learning
  - Adaptive Data Selection
- **Prediction**: Document-level Consistency
  - Dictionary-based
  - Model-based
- **Conclusions**



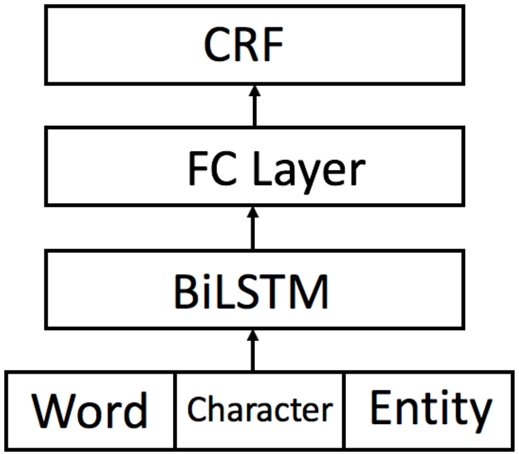
Introduction: Bidirectional LSTM-CRF

- Achieves **state-of-the-art** performance for many sequence labeling tasks
- **Generalize well** due to simple model structure and few parameters
- Very **flexible** architecture, easy to incorporate new ideas
  - Multi-input: include new features
  - Multi-task for transfer learning – natural for hierarchical architecture



Multi-Input Model: Architecture

- Multi-Input model that includes embeddings from
  - word embeddings (GloVe)
  - character embeddings (BiLSTM)
  - **entity embedding**
  - **gazetteer using freebase title**
  - ...
- Entity embeddings
  - Token entity type distribution derived from a Wikipedia Name Tagger (Pan, 2017)
  - Construct embedding by concat such distributions w. additional position features



Multi-Input Model: Entity Embedding

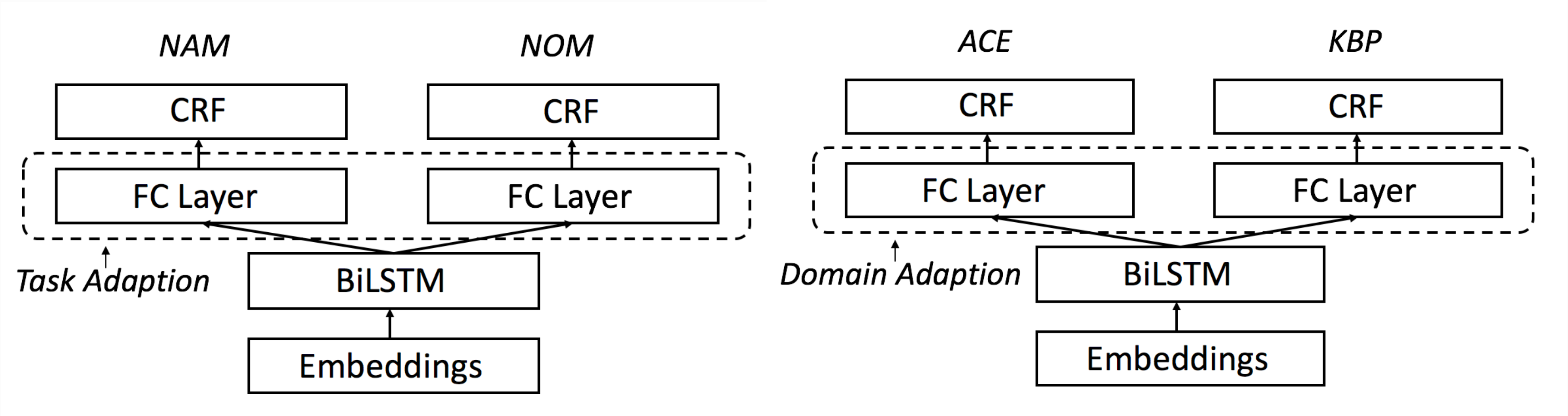
- Entity embedding feature significantly improve the NAM prediction by **3.3 F1 point**
- Freebase feature actually **worsen** the performance
  - Many common words entities
  - Potential improvement with page rank features
- Dictionary constructed from other sources does not help either

Methods	NAM	NOM	Overall
baseline	0.809	0.587	0.748
+ entity embeddings	0.842	0.587	0.770

Table 1: Effectiveness of additional entity embeddings in model embedding layer.

Multi-Task Learning: Architecture

- The hierarchical architecture of BiLSTM-CRF is very natural for **multi-task learning**.
- Bottom components can be **shared** across task/domain.



Multi-Task Learning: Adaptive Data Selection

- Multi-task training can alleviate some of the problem caused by data **heterogeneity** between target and source.
- Data selection algorithm that further **removes noisy data** from source dataset.
- At each iteration, data selection from the source domain is **interleaved** with model parameter updates.
- Training data is selected based on a **consistency score**.

Repeat:

1. Train the model for one iteration, by optimizing the following instance weighted object function,

$$J = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{T}} p(\mathbf{y}|\mathbf{x};\theta^{\mathcal{T}}) + \sum_{(\mathbf{x}',\mathbf{y}')\in\mathcal{S}_{train}} p(\mathbf{y}'|\mathbf{x}';\theta^{\mathcal{S}})$$

2. Compute consistency score for each training example in $\mathcal{S}$,

$$s(\mathbf{x}) = \max_i \sum_j p(x_i = j) \log \frac{p(x_i = j)}{q(x_i = j)},$$

   where $p(x_i) \sim softmax(\phi^{\mathcal{T}}(x_i))$ and $q(x_i) \sim softmax(\phi^{\mathcal{S}}(x_i)))$;

3. Construct $\mathcal{S}_{same}$, $\mathcal{S}_{diff}$ by the following, $\mathcal{S}_{same} = \{\mathbf{x} \in \mathcal{X}^{\mathcal{S}} : s(\mathbf{x}) < \alpha\}$ and $\mathcal{S}_{diff} = \{\mathbf{x} \in \mathcal{X}^{\mathcal{S}} : s(\mathbf{x}) > \beta\}$;
4. Update source training set $\mathcal{S}_{train}$, $\mathcal{S}_{train} \leftarrow \mathcal{S}_{train} \cup \mathcal{S}_{same} \setminus \mathcal{S}_{diff}$.

Until: $|\mathcal{S}_{diff}| < k$

Multi-Task Learning: Experiments

- We use ACE and ERE as source dataset and KBP as target
- MT **does not improve NAM** at all
- MT and data selection **significantly improves NOM**
- Sentences with **plural form nouns** are removed from source, since they are annotated differently from target

Methods	NAM	NOM	Overall
baseline + entity embeddings	0.842	0.587	0.770
+MT	0.841	0.626	0.786
+MT + adaptive data selection	0.842	0.634	0.788

Table 2: Effectiveness of training data consistency.

Doc-level Consistency: Dictionary Based and Model Based

- Observations: NER predictions are **not consistent** across document. E.g. **'Microsoft'** are detected in one sentence but not others; **'MS'** is hard to predict without document level contexts.
- Dictionary-based approach:
  - build a entity dictionary from the predictions in the first pass
  - **expand the dictionary using a KB** (Wikipedia redirect links)
  - match the document with the dictionary in a second pass
- Model-based approach:
  - Build a model that takes predictions of first pass to generate final prediction
  - RNNs suffer **short memory** and **computational expensive**
  - We resorts to use CNN models

ID-CNN (Strubell, 2017)

- CNN
  - **Better memory, faster computation**
- Dilated CNN
  - **context not consecutive**
  - dilated window skips every d inputs
  - Effective context grows exponentially as d grows exponentially
- Iterated Dilated CNN
  - Parameter sharing for stacked DCNN blocks; **avoid overfitting**

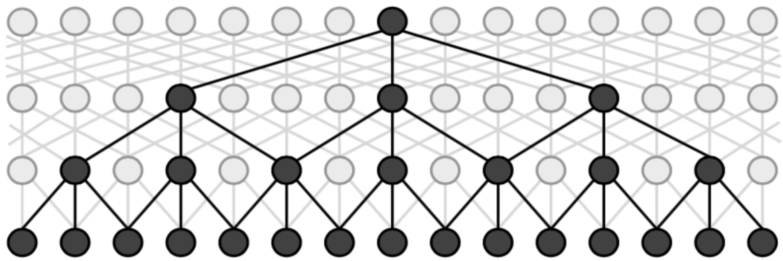


Figure 1: A dilated CNN block with maximum dilation width 4 and filter width 3. Neurons contributing to a single highlighted neuron in the last layer are also highlighted.

Doc-level Consistency: Experiments

- Simple document-level dictionary-based approach **performs as good** as model-based approach on NAM task
  - Corpus-level dictionary **deteriorates** the performance
- Model-based approach **capture additional dependencies** of NOM task
- Future work to combine sentence level and doc level into **single model**

Methods	NAM	NOM	Overall
baseline + entity embeddings	0.842	0.587	0.770
+ label consistency (dictionary based)	0.851	0.587	0.778
+ label consistency (model based)	0.850	0.595	0.779

Table 3: Effectiveness of prediction label consistency.

Final Results with Model Ensemble

- English NERC results for EDL 2016/17
- **1.6 F1 point** improvement with model ensemble
- **0.7 F1 point** improvement with additional training data

Ensemble config	Precision	Recall	F1
Single model	0.833	0.760	0.795
2/4 voting	0.827	0.790	0.808
3/4 voting	0.850	0.776	0.811
Union of two 2/4	0.831	0.791	0.811

Table 4: Overall F1 score with different ensemble configurations.

Year	Our F1	Best F1
2016	0.804	0.772
2017	0.811	0.811

Table 5: Performance comparison between 2016 and 2017 datasets.

Conclusions

- Submitted English name tagging and achieved **F1 0.811-ranking 1st**
- Evaluate and experiment a collection of methods to improve state-of-the-art neural NER model
- External high quality gazetteer works, but **not all-inclusive** ones
- Additional training data works, and **instance selection** further helps
- **Simple doc-level consistency** constraints can work reasonably well

Thanks

Define Smarter
Tomorrow.