Deep Learning for Character-based Information Extraction

Yanjun Qi (University of Virginia, USA)
Sujatha Das G (Penn. State University, USA)
Ronan Collobert (IDIAP)
Jason Weston (Google Research NY)

Task: Target Applications

- Target task: automatically extract information about pre-specified types of events from a linear sequence of unit tokens
  - character-based,
  - no word boundaries,
  - no capitalization cues

- For examples,
  - Chinese language NLP: Chinese-character based
  - Protein sequence tagging: Amino-acid based

Task: Case Study (1):

- Natural Language Processing on Chinese sequences

Characters	克	林	顿	总	统	前	往	中	东
WS	B	I	E	B	E	B	E	B	E
POS	B-NR	I-NR	E-NR	B-NN	E-NN	B-VV	E-VV	B-NR	E-NR
NER	B-PER	I-PER	E-PER	O	O	O	O	B-LOC	E-LOC

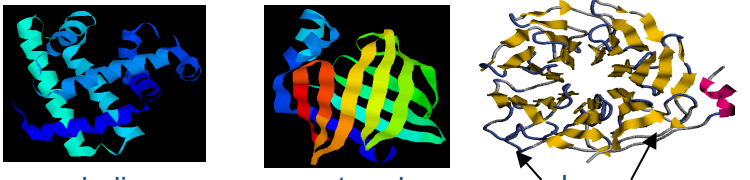
Word Segmentation (WS): Basic task, separate contiguous characters into words
Part of Speech (POS) tagging: Determine part of speech of each word in the text
Name Entity Recognition (NER): determine person, organization and location names in text

Task: Case Study (2):

Protein Sequence ➔ Structural Segments

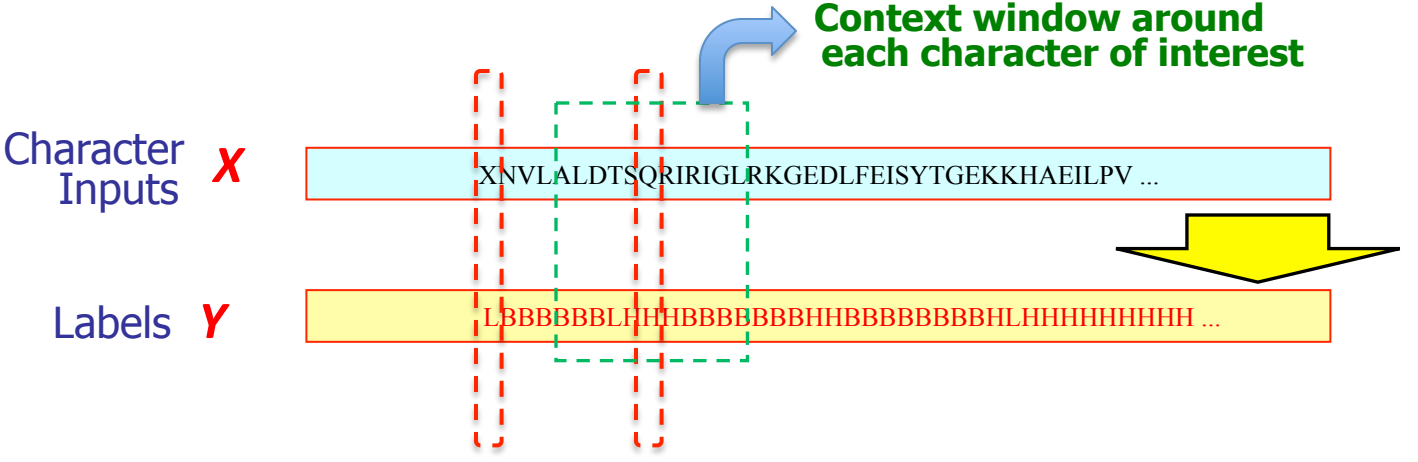
- Input X: Primary sequence

MTYKLILNGKTKGETTTEAVDAATAEKVFQYANDNGVDGEWTYTE



- Output Y: e.g. Secondary structure (SS)

Task: Context-Window based Per-Character Tagging

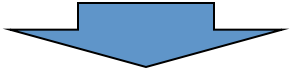


Method: Deep Learning to Rescue

Feature Engineering

✓Most time-consuming in development cycle

✓ Often hand-craft and task dependent in practice



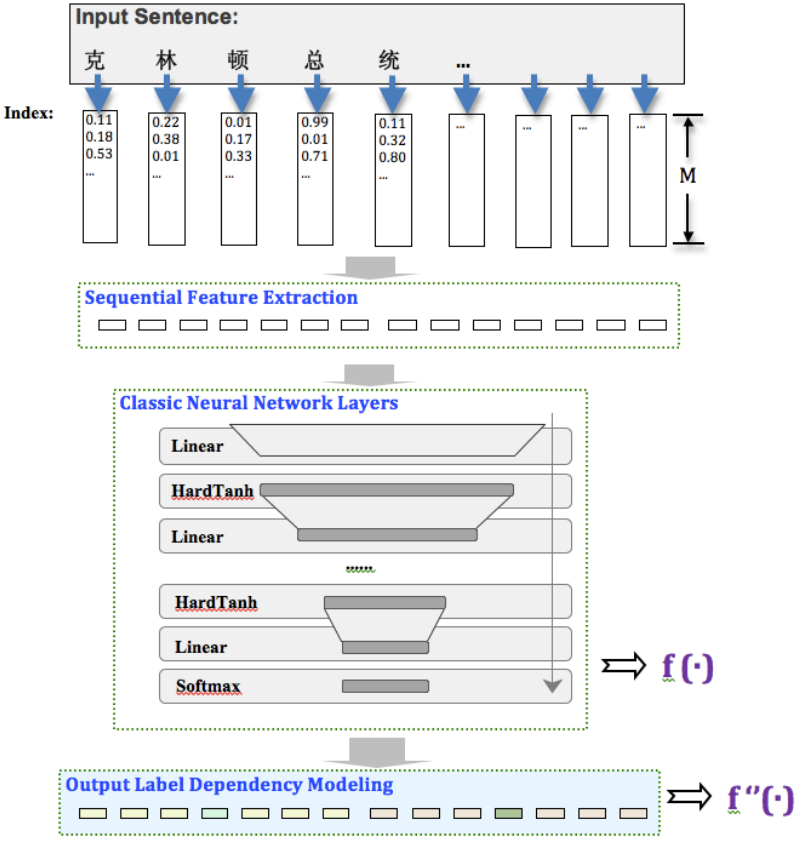
Feature Learning

✓ Easily adaptable to new similar tasks

✓ Layerwise feature representation learning

Previous approaches: Use task-specific/hand-crafted features with a shallow learning structure

Now: Task-independent "deep" structure using simple features input to deep neural network (NN) architecture



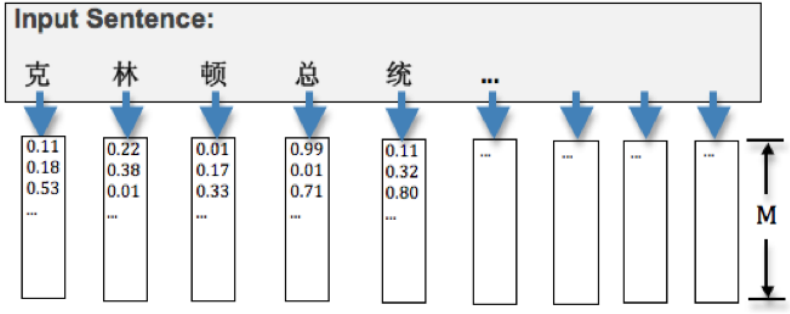
Criteria to train: Negative Log Likelihood
Using Stochastic Gradient descent (SGD)

Learn Feature Representation for each character

Learn Representation for each segment around current position

Learning Function to map from representation to output class label

Method: Character to Vector Representations Learning



- ✓ The first layer in our deep structure;
- ✓ Idea: Characters are embed in a vector space
- ✓ Embedding are trained

How to train this embedding layer:

- ✓ (1). Supervised: Trained as a normal NN layer, using SGD, based on target task's training pairs
- ✓ (2). Initialized with unsupervised "language model" (lm) pre-training: to *captures similarities among characters based on their contexts in the unsupervised sequences*, e.g. Chinese Wiki, swissprot protein sequence DB

Method: Modeling spatial dependency among characters

e.g.

$y_2 - y_2 - y_1 - y_1 - y_1 - y_2 - y_2 - y_3 - y_3 - y_3 - y_2 - y_2$

- A Viterbi algorithm to capture spatial dependencies between y_i
  - i.e. optimize the whole sentence-level log-likelihood
  - i.e. encourage valid paths of output tags
  - = Output tag transition scores + deep network scores

$$\sum_{t=1}^{T} \left([A]_{[i]_{t-1},[i]_t} + [f_\theta]_{[i]_t,t} \right)$$

Experiments : Data Sets

Table 1: Summary of datasets used in experiment

Dataset/Task	#Chars	#UniqueChars	#Sent
POS(CTB)	1,288,840	4,447	28,295
WS(CTB)	1,287,159	4,696	28,295
NER(CITYU)	1,816,417	4,678	43,734
SS (CB513)	83,707	25	497

Table 3: Summary of Output Labels

Task	#Labels	Example Tags
NER	14	B-LOC, I-LOC, S-ORG
POS	107	S-NR, B-NN, E-VV, B-DT
WS	5	B, I, E,S
CB513	4	H, B, C

10

Experiments : Performance Comparison

Configuration / Task	WS-Chinese	POS-Chinese	NER-Chinese	SS-Protein*
1. c1	94.73	86.74	80.61	74.5
2. c1+lm	95.57	86.93	81.79	74.8
3. c1+vit	95.38	88.41	85.81	77.6
4. c1+lm+vit	96.07	88.81	86.99	77.8
5. c1+lm+c2	95.98	88.48	83.51	~
6. c1+lm+c2+vit	**96.62**	89.39	87.24	~
7. c1+lm+c2+vit+ws	~	**93.27**	88.88	**80.3***
Previous Best	95.9 [10]	91.9 [10]	**89.00** [6]	80.0 [5]
Previous Second Best	95.1 [10]	91.3 [10]	88.61 [6]	~

c1: character unigrams, c2: character bigrams,
lm: embedding obtained with deep language model,
vit: Viterbi algorithm

Summary:

Why is our method preferable?

- No particular task-specific feature engineering.
- Robust and flexible
- Easily adaptable to other character-based tagging tasks, e.g. Japanese NER

References

[0] http://www.cs.cmu.edu/~qyj/zhSenna/
[1] R. Collobert et al, Natural language processing (almost) from scratch, JMLR 12
[2] Qi et al, A unified multitask architecture for predicting local protein properties. PLoS ONE 12
[3] Levow, G.A.: The third international chinese language processing bakeoff, SIGHAN 2006
[4] Y. LeCun et al. 1998. Efficient BackProp.