Deep Learning for Chinese Word Segmentation and POS Tagging

Group 6

Hong Zhu[1], Guoxing Yao[2], Wanyue Xiao[1], Zequn Che[1] Instructor: Prof. Lu Xiao
School of Information Studies[1] & School of Engineering[2], Syracuse University, Syracuse, NY 13244 USA

Abstract

Word Segmentation and Part-Of-Speech tagging are two preliminary works for Chinese text processing. Various studies had been conducted, such as conditional Random Fields (CRF), and extraordinary achievements have been achieved.

Still, there are **four major shortcomings**:
- Large volume of data
- Over-fitting issue
- Time-consuming
- Intractable decoding

Multiple Neural Networks

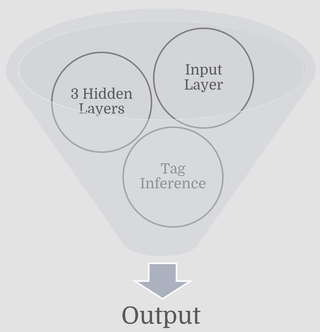
Build a Perceptron-style Algorithm (PSA)

Achieve close to state-of-the-art performance

Neural Network Mechanism

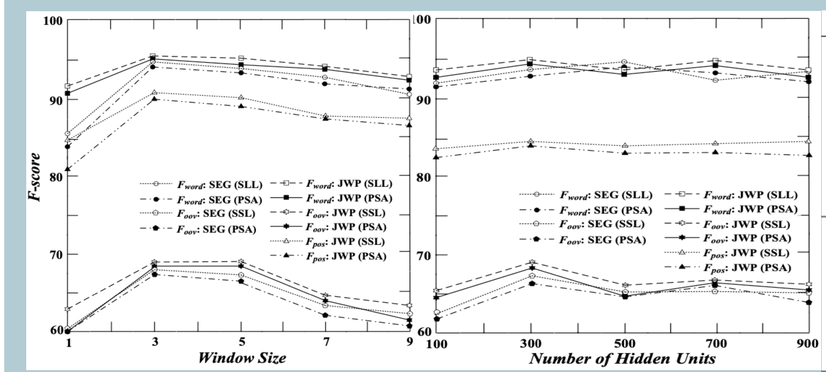
Choose to use variant of the neural network architecture for probability language model

Working Steps:
Get essential features from window of characters
Establish neural network layers
Utilize Viterbi algorithm to generate result that contains tag inference.



Experiments Result

	Dataset	Objective	Result
1st	Chinese Treebank 4	Hyper-parameter	Setting window size as 3 or 5 & setting number of hidden layers unit to 300
2nd	Chinese Treebank from Bakeoff-3	Performance on Labeled Data	Having a recall value of 92.59 and 93.83 in SEG and JWP respectively
3rd	325MB unlabeled SINA News	Performance on Unlabeled Data	Having a recall value of 94.57 and 95.23 in SEG and JWP respectively



	Approach	F_{word}	R_{oov}	F_{pos}
SEG	(Zhao et al., 2006)	93.30	70.70	–
	(Wang et al., 2006)	93.00	68.30	–
	(Zhu et al., 2006)	92.70	63.40	–
	(Zhang et al., 2006)	92.60	61.70	–
	(Feng et al., 2006)	91.70	68.00	–
	PSA	92.59	64.24	–
	PSA + LM	94.57	70.12	–
JWP	(Ng and Lou, 2004)	95.20	–	–
	(Zhang and Clark, 2008)	95.90	–	91.34
	(Jiang et al., 2008)	97.30	–	92.50
	(Kruengkrai et al., 2009)	96.11	–	90.85
	PSA	93.83	68.21	90.79
	PSA + LM	95.23	72.38	91.82

Perceptron-style Algorithm

Input:

$\mathcal{R}$: a training set.
N: a specified maximum number of iterations.
E: a desired tagging precision.

Initialization: set the initial parameters of the network with small random values.

Output: the trained parameters $\tilde{\theta}$

Algorithm:

do

for each example $(c, t) \in \mathcal{R}$

get the matrix $f_\theta(c)$ by the neural network under the current parameters θ

find the highest scoring sequence of tags t' for c with $f_\theta(c)$ and A_{ij} by using the Viterbi algorithm

if $(t \neq t')$

compute the gradients with respect to $f_\theta(c)$ and A_{ij} as (11) and (12)

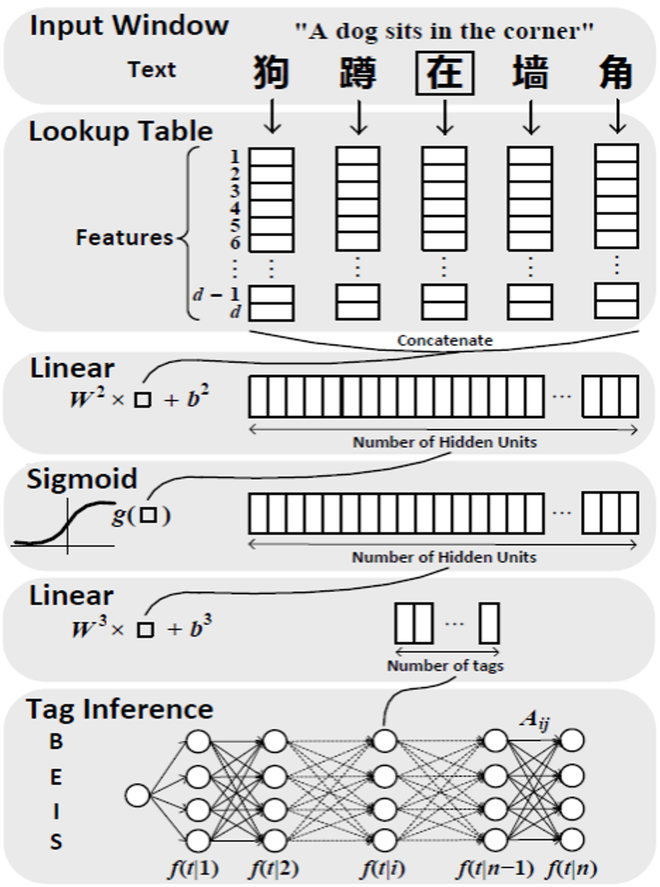
compute the gradients with respect to the weights from output layer to character embedding layer

update the parameters of the network by (9)

until the desired precision E achieved or maximum number of iterations N reached

return $\tilde{\theta}$

Architecture



Process

Mapping Characters into Feature Vectors
Lookup Table ⇥ Matrix
Column ⇥ Vector of dimension of d
D ⇥ dictionary from training set |D|

Tag Scoring
A neural network can be considered as a function $f_\theta(\cdot)$ with parameter θ. Any feed-forward neural network with L layers can ben seen as a composition of functions $f_\theta^l(\cdot)$ defined for each layer l:

$$f_\theta(\cdot) = f_\theta^l(f_\theta^{l-1}(\dots f_\theta^1(\cdot)\dots))$$

Tag Inference
Given an input sentence $c_{[1:n]}$, the network outputs the matrix of scores $f_\theta(t|i)$ to indicate the score output by the network with θ at the i-th character.
The score of a sentence $c_{[1:n]}$ along a path of tags $t_{[1:n]}$ is then given by the sum of transition and network scores.

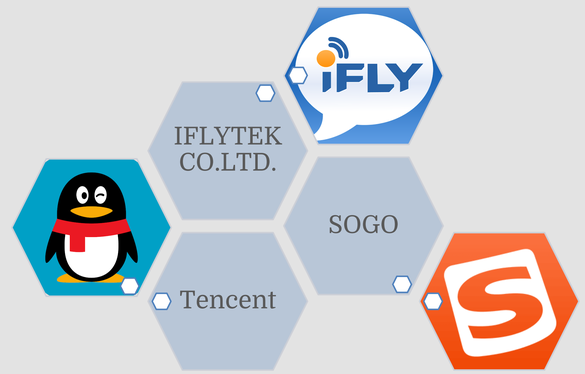
$$s(c_{[1:n]}, t_{[1:n]}, \theta) = \sum_{i=1}^{n}(A_{t_{i-1}t_i} + f_\theta(t_i|i))$$

Two Methods:
- Sentence-Level Log-Likelihood (SLL)
- Perceptron-style Algorithm (PSA)

Conclusion

Advantages of PSA:
- Easier to implement
- Much faster
- Losing of performance is negligible

Real World Scenarios



References

- Forney, G.D., 1973, The Viterbi Algorithm. *Processing of the IEEE*, 61(3), pp.268-278.
- Zhang, X., Chen, H. and Xu, T., 2013, Deep learning for Chinese word segmentation and POS tagging, *Empirical Methods in Natural Language Processing*, pp.647-657.
- Collobert, R. 2011, Deep learning for efficient discriminative parsing, *In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS'11)*