Neural Word Segmentation Learning for Chinese

Deng Cai and Hai Zhao

Shanghai Jiao Tong University

thisisjcykcd@gmail.com

August 8, 2016

Overview

1 Motivation
- Task Introduction
- Previous Methods
- Task Review

2 Neural Word Segmentation Learning
- Overview
- Neural Scoring Model
- Beam Search

3 Experiments
- Model Analysis
- Comparison with Prior Methods

Chinese Word Segmentation

Most east Asian languages including Chinese are written without explicit word delimiters.

As word is recognized as the fundamental unit for most NLP tasks, word segmentation is a preliminary step for processing those languages.

Main challenges

- Ambiguity
- Out-of-vocabulary words

Previous Methods

- Character based methods (sequence labeling)
- Word based methods

Sequence Labeling

Sequence labeling has been the standard approach to Chinese word segmentation since (Xue, 2003) (dominated this field for 13 years).

However, people do not tag individual characters when they are reading Chinese. Sequence labeling is effective in computational linguistics but not quite natural for linguistic cognition.

Sequence Labeling

Other drawbacks inside sequence labeling schemes include

- Tag-tag transition is insufficient to model the complete influence from historical decisions.
- Fixed sized window restricts the flexibility of capturing useful information at diverse distances.
- Word-level information is unemployed.

Word-based Methods

Most of them follow the spirit in (Zhang and Clark, 2007).

Previous word-based methods are restricted by.

- Manual effort in feature engineering.
- Word interacting can not be fully modeled.

Task Review

The ultimate goal of word segmentation algorithms is to output a word sequence (i.e, sentence) that satisfies the following two requirements when given a character sequence.

Legal word

YES: 飞机 (airplane)/场在 (ILLEGAL)/维修 (repair)
NO: 飞机场 (airport)/在 (is under)/维修 (repair)

Natural sentence (complete, coherent and smooth)

NO: 勇敢 (boldness)/的士 (taxi)/兵 (soldier)
YES: 勇敢的 (brave)/士兵 (soldier)

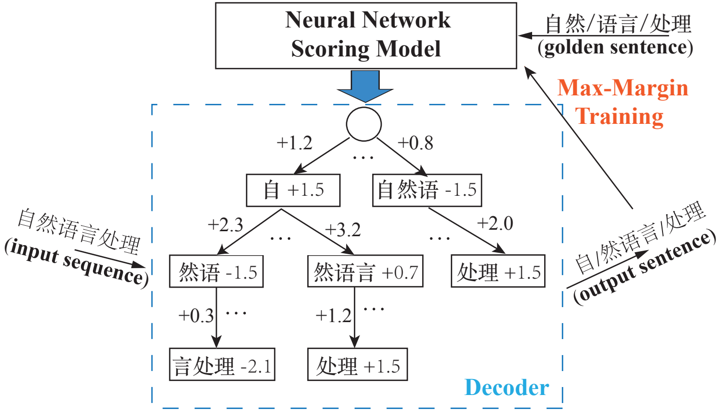
Formalization

Given input character sequence x, output sentence y^*,

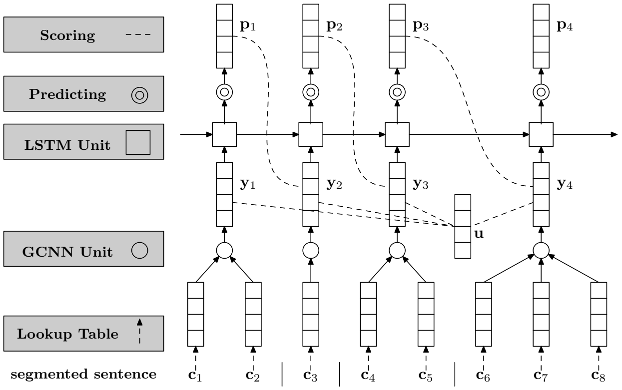
$$y^* = \arg\max_{y \in \mathsf{GEN}(x)} \left(\sum_{i=1}^{n} \mathsf{score}(y_i | y_1, \cdots, y_{i-1}) \right)$$

where $\mathsf{GEN}(x)$ denotes the set of all possible segmentations for the input sequence x.

Framework


Neural Network Scoring Model
自然/语言/处理
(golden sentence)
Max-Margin Training
+1.2
+0.8
自 +1.5
自然语 -1.5
+2.3
+3.2
+2.0
自然语言处理
(input sequence)
然语 -1.5
然语言 +0.7
处理 +1.5
自/然语言/处理
(output sentence)
+0.3
+1.2
言处理 -2.1
处理 +1.5
Decoder

Model Architecture



Benefits

Models		Characters	Words	Tags
character based	(Zheng et al., 2013), ...	$c_{i-2}, c_{i-1}, c_i, c_{i+1}, c_{i+2}$	-	$t_{i-1}t_i$
	(Chen et al., 2015b)	$c_0, c_1, \ldots, c_i, c_{i+1}, c_{i+2}$	-	$t_{i-1}t_i$
word based	(Zhang and Clark, 2007), ...	c in w_{j-1}, w_j, w_{j+1}	w_{j-1}, w_j, w_{j+1}	-
	Ours	$c_0, c_1, \ldots, c_i$	$w_0, w_1, \ldots, w_j$	-

- Model the segmentation structure straightforward.
- Cover information at all levels (character, word and sentence).
- Make use of complete historical information (both plain text and decisions)
  - No sliding window is adapted.
  - No Markov assumption is made.

Beam Search

Problem

The total number of possible segmentations grows **exponentially** with the length of input sequence.

Solution

Split segmentation into two parts, (i) the last word, (ii) the sub segmentation in front of (i).

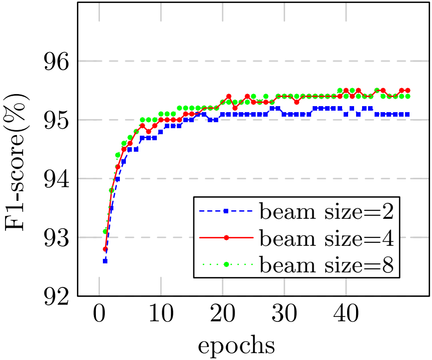
Approximate k-best segmentations of its prefixes iteratively.

Input: model parameters θ
beam size k
maximum word length w
input character sequence $c[1:n]$
Output: Approx. k best segmentations

$\pi[0] \leftarrow \{(score = 0, \mathbf{h} = \mathbf{h}_0, \mathbf{c} = \mathbf{c}_0)\}$
for $i = 1$ to n **do**
  ▷ Generate Candidate Word Vectors
  $X \leftarrow \emptyset$
  for $j = \max(1, i - w)$ to i **do**
    $\mathbf{w} = \text{GCNN-Procedure}(c[j:i])$
    $X.\text{add}((index = j - 1, word = \mathbf{w}))$
  end for
  ▷ Join Segmentation
  $Y \leftarrow \{\ y.\text{append}(x) \mid y \in \pi[x.index]$ and $x \in X\}$
  ▷ Filter k-Max
  $\pi[i] \leftarrow k\text{-}\arg\max_{y \in Y} y.score$
end for
return $\pi[n]$

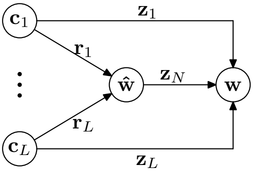
Beam Size

Performances of different beam sizes on PKU dataset.



Good balance between accuracy and efficiency.

Gated Combination Neural Network (GCNN)

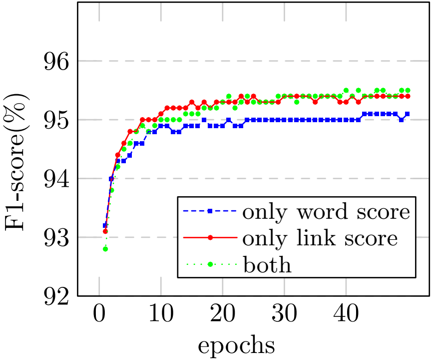


Performances of different models on PKU dataset.

models	P	R	F
Single layer ($d = 50$)	94.3	93.7	94.0
GCNN ($d = 50$)	95.8	95.2	95.5
Single layer ($d = 100$)	94.9	94.4	94.7

Link Score & Word Score

Performances of different score strategies on PKU dataset.



Link score plays a critical role in gaining performance improvement.

Comparison with Prior Neural Models

Results with * are from our runs on their released implementations.

Models	PKU			MSR		
	P	R	F	P	R	F
(Zheng et al., 2013)	92.8	92.0	92.4	92.9	93.6	93.3
(Pei et al., 2014)	93.7	93.4	93.5	94.6	94.2	94.4
(Chen et al., 2015a)*	94.6	94.2	94.4	94.6	95.6	95.1
(Chen et al., 2015b)*	94.6	94.0	94.3	94.5	95.5	95.0
This work	**95.5**	**94.9**	**95.2**	**96.1**	**96.7**	**96.4**
+Pre-trained character embedding						
(Zheng et al., 2013)	93.5	92.2	92.8	94.2	93.7	93.9
(Pei et al., 2014)	94.4	93.6	94.0	95.2	94.6	94.9
(Chen et al., 2015a)*	94.8	94.1	94.5	94.9	95.9	95.4
(Chen et al., 2015b)*	95.1	94.4	94.8	95.1	96.2	95.6
This work	**95.8**	**95.2**	**95.5**	**96.3**	**96.8**	**96.5**

Comparison with State-of-the-Art Models

Results with * used external dictionary or corpus.

Models	PKU	MSR	PKU	MSR
(Tseng et al., 2005)	95.0	96.4	-	-
(Zhang and Clark, 2007)	94.5	97.2	-	-
(Zhao and Kit, 2008b)	95.4	97.6	-	-
(Sun et al., 2009)	95.2	97.3	-	-
(Sun et al., 2012)	95.4	97.4	-	-
(Zhang et al., 2013)	-	-	96.1*	97.4*
(Chen et al., 2015a)	94.5	95.4	96.4*	97.6*
(Chen et al., 2015b)	94.8	95.6	96.5*	97.4*
This work	**95.5**	96.5	-	-

Results Analysis

Long words (with length > 4) account for 0.19% in PKU test set but 1.07% in MSR test set.

Max. word length	F_1 score	Time (Days)
4	96.5	4
5	96.7	5
6	96.8	6

Words with very large (> 6) lengths still account for 0.42% in MSR test set.

Problems with longer words

- less training data (most of them are hierarchical entity names).
- more parameters to train (GCNN part).

Questions are welcome!

E-mail:thisisjcykcd@gmail.com