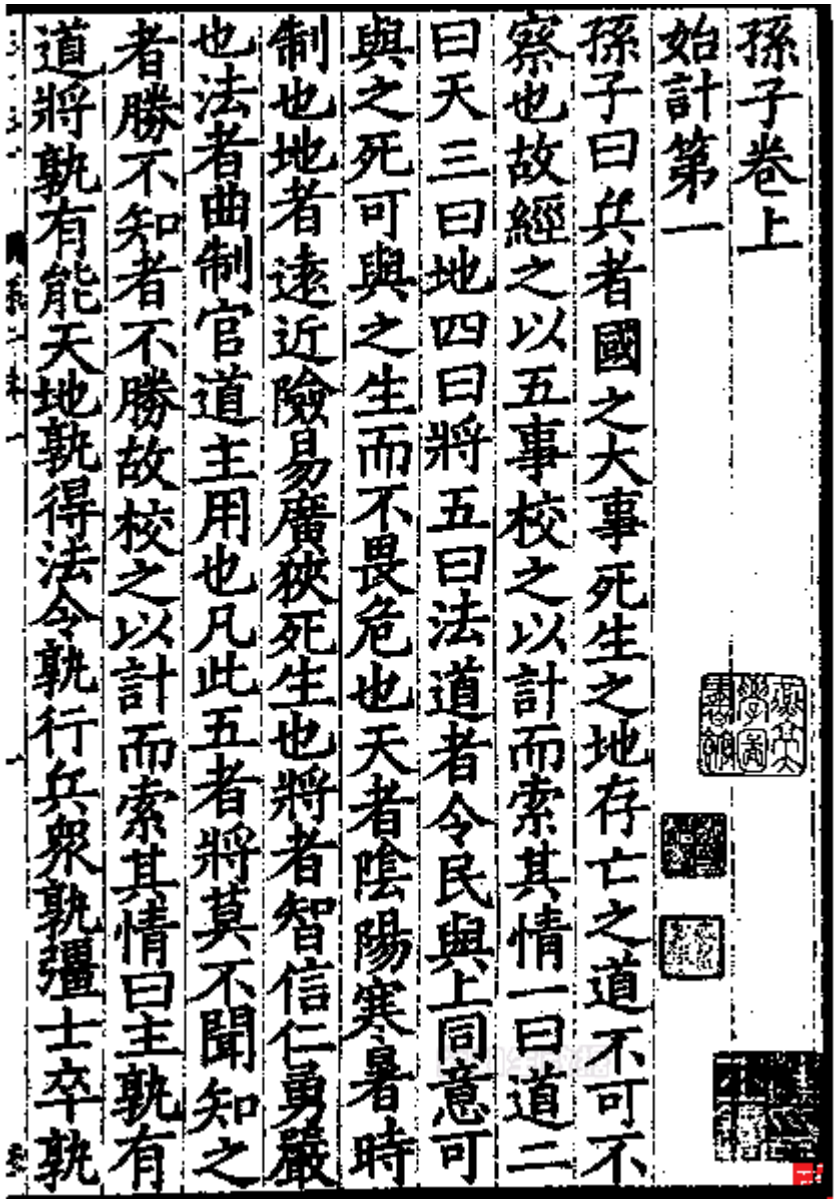
一种基于神经网络语言模型的古文断句方法

王博立 史晓东† 谭知行 陈毅东 王伟莉
厦门大学智能科学与技术系

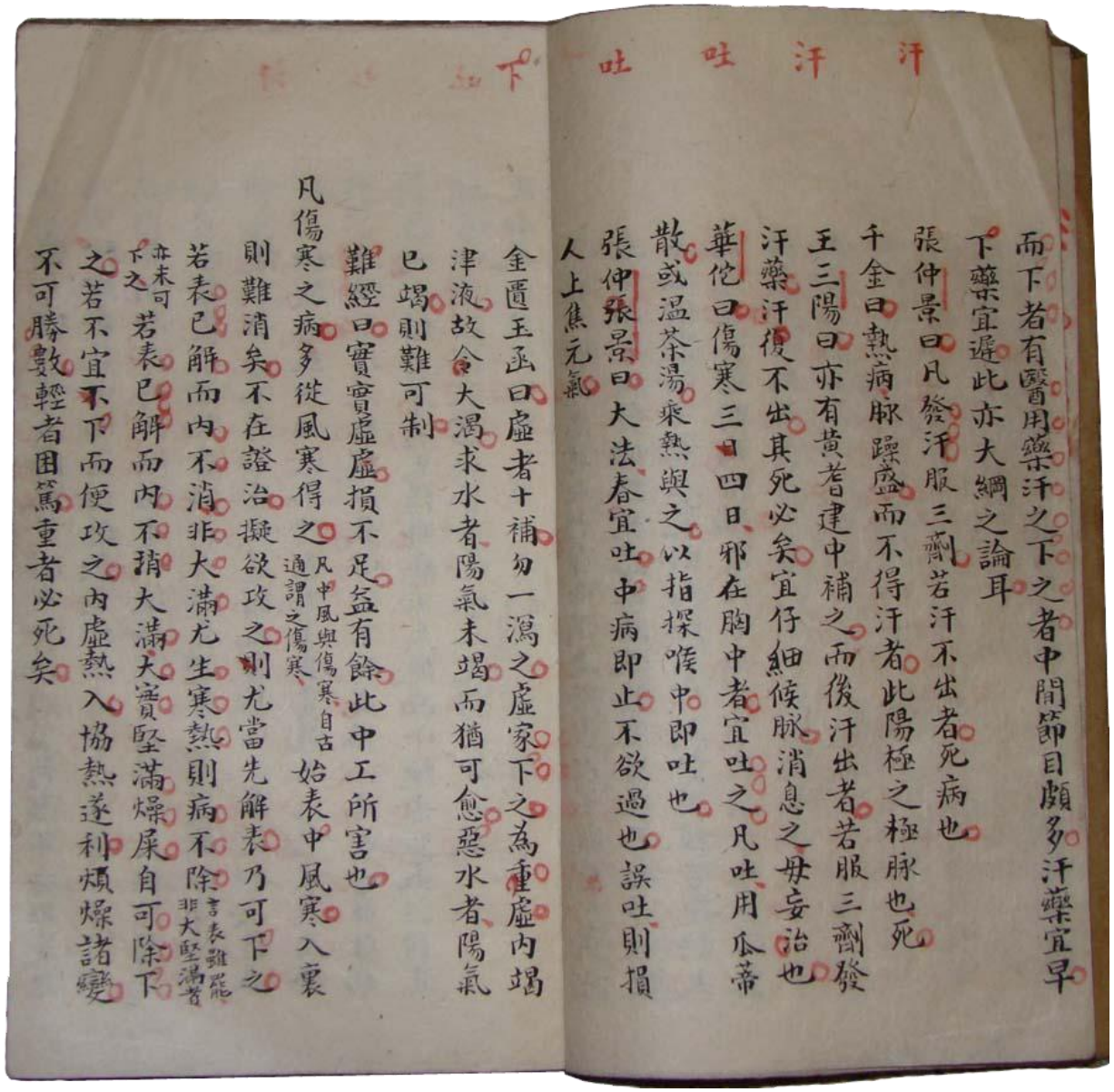
Introduction

- 古人著书不加标点断句


孫子卷上
始計第一
孫子曰兵者國之大事死生之地存亡之道不可不
察也故經之以五事校之以計而索其情一曰道二
曰天三曰地四曰將五曰法道者令民與上同意可
與之死可與之生而不畏危也天者陰陽寒暑時
制也地者遠近險易廣狹死生也將者智信仁勇嚴
也法者曲制官道主用也凡此五者將莫不聞知之
者勝不知者不勝故校之以計而索其情曰主孰有
道將孰有能天地孰得法令孰行兵衆孰彊士卒孰

Introduction

- 古人著书不加标点断句
- 读者自行断句：句读



Introduction

- 古人著书不加标点断句
- 读者自行断句：句读
- 古文自动断句技术
  - 大部分传世古籍尚未断句
  - 断句能方便现代人阅读和理解古籍
  - 断句是进一步古籍处理（如分词等）的必要前序步骤

Introduction

- 已有研究：基于CRF的断句模型
  - （张开旭，2009）
  - （张合，2009）
  - （Hen-Hsen Huang，2010）
- 不足
  - 手工设计特征模板
  - 特征数量有限
  - 未利用语义信息
  - 数据集规模小

Our Model

- 将古文断句看作一个序列生成问题

$$\hat{Y} = \underset{Y, Y \in U(X)}{\operatorname{argmax}} P(Y)$$

  - X ：输入的未断句的文本
  - Y ：添加了标点的文本
  - $P(Y)$ ： Y的概率得分
  - $U(X)$ ： X所有可能的断句结果

    U(“三人行”) = {“三人行”, “三。人行”, “三人。行”, “三。人。行”}

Our Model

- 模型1：语言模型

$$P(Y) = P_{lm}(Y)$$

  - 不足：语言模型倾向于短文本=>添加标点少=>recall低

- 模型2：语言模型 + 长度惩罚

$$P(Y) = P_{lm}(Y) + \lambda Len(Y)$$

  - λ为长度惩罚系数
  - $P_{lm}(Y)$调整为概率对数

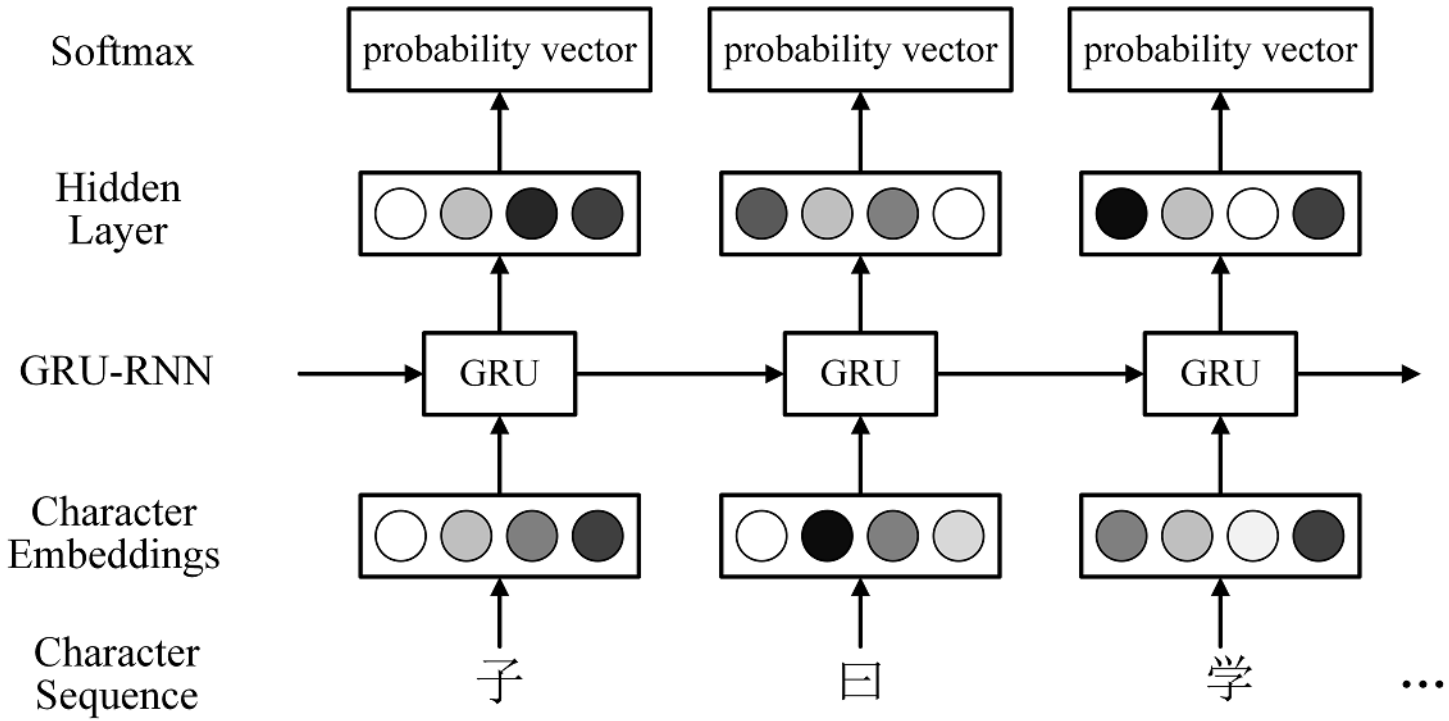
Our Model

- 解码：
  - Beam Search（实验中beam的大小取15）

- 标题的处理：启发式规则
  - 若$\hat{Y}$中含有断句标记，则在$\hat{Y}$末尾添加标点（段落）
  - 若$\hat{Y}$中不含断句标记，则$\hat{Y}$末尾不添加标点（标题）

Our Model

- 语言模型
  - 基于GRU的循环神经网络语言模型（GRU-RNNLM）



Our Model

- RNNLM的条件概率建模

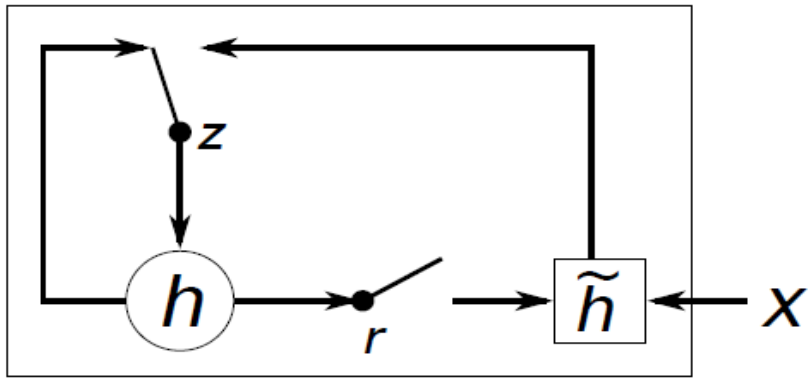
$$P(w_{t+1} = i | w_t \cdots w_1) = \frac{exp(\boldsymbol{W}_i \boldsymbol{h}_t)}{\sum_{j=1}^{K} exp(\boldsymbol{W}_j \boldsymbol{h}_t)}$$

- 隐状态$\boldsymbol{h}_t$
  - 由循环神经网络编码的上下文语义信息
  - 更新： $\boldsymbol{h}_t = f(\boldsymbol{x}_t, \boldsymbol{h}_{t-1})$
  - 我们使用GRU（Gated Recurrent Unit）来表示f

Our Model

- GRU（Gated Recurrent Unit）
  （Chung，2014；Cho，2014）

$$\boldsymbol{r} = sigmoid(\boldsymbol{W}_r \boldsymbol{x}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1})$$
$$\boldsymbol{z} = sigmoid(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1})$$
$$\tilde{\boldsymbol{h}} = tanh(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}(\boldsymbol{r} \circ \boldsymbol{h}_{t-1}))$$
$$\boldsymbol{h} = \boldsymbol{z} \circ \boldsymbol{h}_{t-1} + (1 - \boldsymbol{z}) \circ \boldsymbol{h}_t$$



Experiments

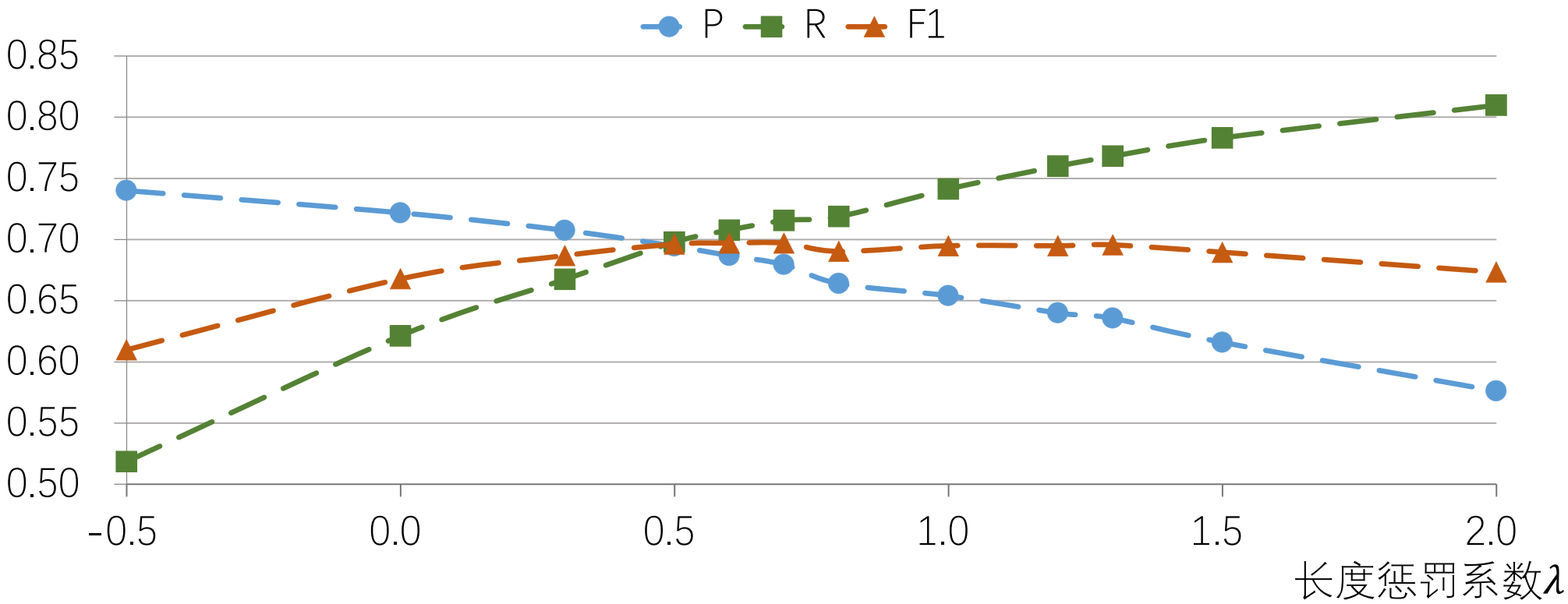
- 语料
  - 至善繁体汉语语料库

    http://cloudtranslation.cc/corpus_tc.html

数据集	规模	字表大小	内容	来源
训练集	2.37 亿字	23905	“至善语料库”中的繁体古籍部分	至善
开发集	1 万字	1890	《阅微草堂笔记》（卷一）	中研院
测试集 1	32 万字	6188	《宾退录》《朝野佥载》《南部新书》《楚辞补注》《中吴纪闻》	是何年
测试集 2	36 万字	5755	《阅微草堂笔记》（卷二至卷二十四） 《敬斋古今黈》	中研院 是何年

Experiments

- 长度惩罚实验
  - 实验结果：λ=0.65时F1最高



Experiments

- 模型对比实验
  - 模型
    - CRF：（张合，2009）基于CRF的断句模型（CRF++重现）
    - CLM6：保留6种标点（。/？/！/，/；/：）的RNNLM
    - CLM1：把6个标点统一为断句标记（“。”）的RNNLM
  - 实验结果：与CRF可比

	测试集 1			测试集 2		
	P	R	F1	P	R	F1
CLM6 (λ =0)	0.7822	0.5767	0.6639	0.7283	0.6364	0.6793
CLM6 (λ =0.65)	0.7492	0.6727	0.7089	0.6861	0.7205	0.7029
CLM1 (λ =0.65)	0.7212	**0.7325**	0.7268	0.6393	**0.7691**	0.6982
CRF	**0.8163**	0.6617	**0.7309**	**0.7856**	0.7100	**0.7459**

Experiments

- 使用大规模简体语料的实验
  - 已断句简体古籍规模超过繁体
  - 简体汉字数量少于繁体汉字
  - 训练语料：6.29亿字简体+2.37亿字繁体转简体
  - 模型：CLM1-S（与CLM1相同方法训练）
  - 实验结果：大幅提升断句性能

	测试集 1			测试集 2		
	P	R	F1	P	R	F1
CLM1 (λ =0.65)	0. 7212	0. 7325	0. 7268	0. 6393	0. 7691	0. 6982
CLM1-S (λ =0.65)	**0. 8243**	**0. 7988**	**0. 8113**	**0. 7180**	**0. 8339**	**0. 7716**

Experiments

- 按年代划分古籍语料的实验
  - 上古汉语：先秦至西汉
  - 中古汉语：东汉魏晋南北朝
  - 近代汉语：唐五代以后

Experiments

- 按年代划分古籍语料的实验
  - 测试集1、测试集2均来自近代汉语
  - 测试集3：测试集1中去除《楚辞补注》
  - 训练语料：从2.37亿字中抽取近代汉语1.14亿字
  - 模型：CLM1-J （与CLM1相同方法训练）
  - 实验结果：更小更同构的训练集=>更好的断句性能

	测试集 1			测试集 2			测试集 3		
	P	R	F1	P	R	F1	P	R	F1
CLM1 (λ =0.65)	**0. 7212**	0. 7325	**0. 7268**	0. 6393	0. 7691	0. 6982	**0. 6496**	0. 7478	0. 6952
CLM1-J (λ =0.65)	0. 6659	**0. 7544**	0. 7074	**0. 6442**	**0. 8267**	**0. 7241**	0. 6387	**0. 8076**	**0. 7133**

Summary

- 采用基于GRU-RNNLM的序列生成模型能有效对古文进行断句，达到与传统CRF模型可比的效果

- 有效提高断句性能的方法
  - 引入长度惩罚
  - 采用更大规模的简体训练语料
  - 训练不同年代的子模型

- 下一步工作
  - 将神经网络语言模型与传统CRF模型相结合
  - 进行古籍异体字规范化以提高断句性能

Our Latest Work

- 基于GRU的双向循环神经网络断句模型
  - 基于GRU-RNN的seq-2-seq序列标注
  - 大幅提高断句效果：超过CRF模型

模型	测试集 1			测试集 2		
	P	R	F1	P	R	F1
CLM1+0.65LP	0.7212	0.7325	0.7268	0.6393	**0.7691**	0.6982
CRF	**0.8163**	0.6617	0.7309	**0.7856**	0.7100	0.7459
GRU-RNN+1.4ST+0.3LP(beam=200)	0.7629	**0.7475**	**0.7551**	0.7381	0.7564	**0.7472**

  - 断句速度大幅提高

模型	时间（ms）
CLM1+0.65LP（CPU）	274860
GRU-RNN+1.4ST+0.3LP(beam=200)（GPU）	46713

Thanks.

Presented by Boli Wang

2016-05-20