

学术报告中的一些设计技巧

报告人：刘一佳

导师：秦兵、车万翔

错误地利用 报告与论文结构的相似性

Challenges and Contribution

- The first challenge is deriving an optimal alignment in ambiguous situations.
- The second challenge is deriving semantically matched word pairs during the alignment process.
- The third challenge is deriving the rule-based and unsupervised aligner by using the alignment with downstream parser training.
- We proposed an enhanced aligner tuned by transition-based oracle parser.

简介

Overview



Our aligner algorithm

- Enhancing aligner with rich semantic resources
- Producing more accurate alignments

模型



Our oracle parser

模型

Experiments

- We conduct experiments on LDC2014T12
- We evaluate the alignment F1 score and Smatch of resulted parser

实验

Aligner	Aligner	Aligner	Aligner	Aligner	Aligner
Aligner	Aligner	Aligner	Aligner	Aligner	Aligner
Aligner	Aligner	Aligner	Aligner	Aligner	Aligner
Aligner	Aligner	Aligner	Aligner	Aligner	Aligner
Aligner	Aligner	Aligner	Aligner	Aligner	Aligner
Aligner	Aligner	Aligner	Aligner	Aligner	Aligner
Aligner	Aligner	Aligner	Aligner	Aligner	Aligner
Aligner	Aligner	Aligner	Aligner	Aligner	Aligner
Aligner	Aligner	Aligner	Aligner	Aligner	Aligner
Aligner	Aligner	Aligner	Aligner	Aligner	Aligner

Conclusion

- We propose a new MRL aligner which is tuned by a novel transition-based MRL oracle parser. Our aligner is also enhanced with semantic resources and results more alignments.
- Both the word pairs and alignments show the effectiveness of our aligner by using higher alignment F1 score and Smatch score, improving two open-source MRL parsers.
- We also develop transition-based MRL parser based on our aligner and transition system, and it achieves a performance of 88.4 Smatch F1 score via ensemble with only words and POS tags as input.

结论

思考题

- 为什么做学术报告
 - 为了更好地交流
- 做怎样的学术报告
 - ☐ “向听众展示我对问题的深入理解”
 - ☐ “让听众明白我的论文中的技术”
 - ☐ “引起听众的兴趣”

思考题

- 为什么做学术报告
 - 为了更好地交流
- 做怎样的学术报告
 - ☐ “向听众展示我对问题的深入理解”
 - ☐ “让听众明白我的论文中的技术”
 - ☒ “引起听众的兴趣”

听众模型

理想中的听众

- 领域专家
- 已经读过你的论文
- 对于你的工作非常感兴趣

现实中的听众

- 来自其他领域
- 刚刚了解到你的工作
- 这个时段没什么可听的，恰巧发现这屋子网络比较好

类比审稿人模型

审稿

你以为审稿人应该是这样审稿的：

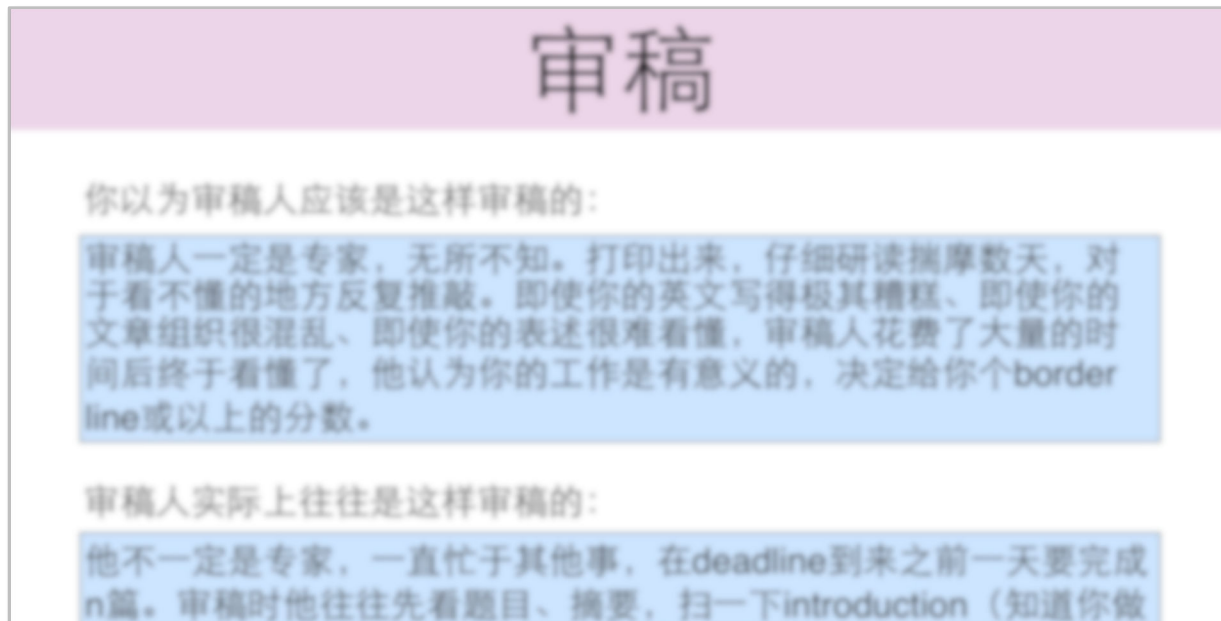
审稿人一定是专家，无所不知。打印出来，仔细研读揣摩数天，对于看不懂的地方反复推敲。即使你的英文写得极其糟糕、即使你的文章组织很混乱、即使你的表述很难看懂，审稿人花费了大量的时间后终于看懂了，他认为你的工作是有意义的，决定给你个border line或以上的分数。

审稿人实际上往往是这样审稿的：

他不一定是专家，一直忙于其他事，在deadline到来之前一天要完成n篇。审稿时他往往先看题目、摘要，扫一下introduction（知道你做什么），然后直接翻到最后找核心实验结果（做得好不好），然后基本确定录还是不录（也许只用5分钟！）。如果决定录，剩下就是写些赞美的话，指出些次要的小毛病。如果决定拒，下面的过程就是细看中间部分找理由拒了。

第一印象定录拒，5分钟内打动审稿人

类比审稿人模型



“You have **two minutes** to engage your audience before they start to doze.” -- Simon Peyton Jones in *How to give a great research talk*

简介部分：展示最好的部分

(Zhang and Nivre 2011, Martins et al 2013)



Our Work

- A neural network based dependency parser!

Parsing on English Penn Treebank (§23):

		Unlabeled attachment score (UAS)	sent / s
Transition -based	MaltParser (greedy)	89.9	560
	Our Parser (greedy)	92.0	1013
	Zpar: beam = 64	92.9*	29*
Graph -based	MSTParser	92.0	12
	TurboParser	93.1*	31*

A Fast and Accurate Dependency Parser using Neural Networks

3

模型部分：多用例子



EM with Features

EM

Fit Params

Initialize weights w

repeat

- Compute expected counts e

repeat

- Compute $\ell(w, e)$
- Compute $\nabla \ell(w, e)$
- $w \leftarrow \text{climb}(w, \ell(w, e), \nabla \ell(w, e))$

until convergence

- Transform w to θ

until convergence



EM with Features

Initialize weights w

repeat

- Compute expected counts e

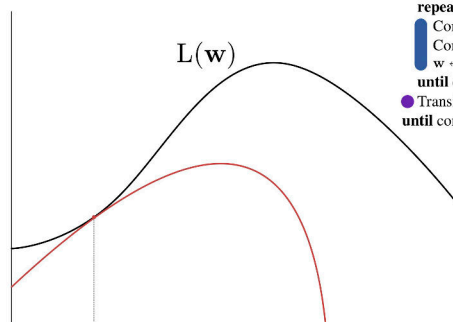
repeat

- Compute $\ell(w, e)$
- Compute $\nabla \ell(w, e)$
- $w \leftarrow \text{climb}(w, \ell(w, e), \nabla \ell(w, e))$

until convergence

- Transform w to θ

until convergence



EM with Features

Initialize weights w

repeat

- Compute expected counts e

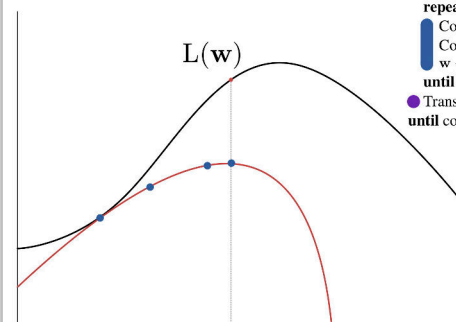
repeat

- Compute $\ell(w, e)$
- Compute $\nabla \ell(w, e)$
- $w \leftarrow \text{climb}(w, \ell(w, e), \nabla \ell(w, e))$

until convergence

- Transform w to θ

until convergence



EM with Features

Initialize weights w

repeat

- Compute expected counts e

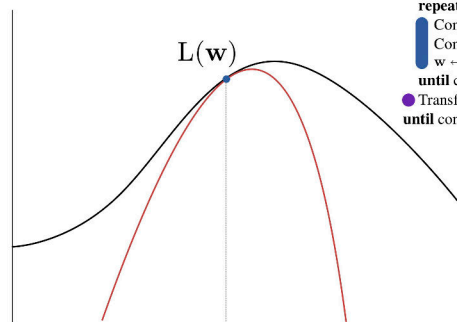
repeat

- Compute $\ell(w, e)$
- Compute $\nabla \ell(w, e)$
- $w \leftarrow \text{climb}(w, \ell(w, e), \nabla \ell(w, e))$

until convergence

- Transform w to θ

until convergence



EM with Features

Initialize weights w

repeat

- Compute expected counts e

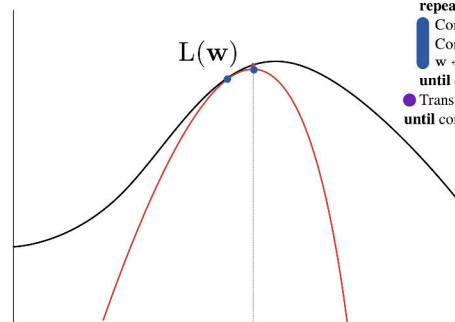
repeat

- Compute $\ell(w, e)$
- Compute $\nabla \ell(w, e)$
- $w \leftarrow \text{climb}(w, \ell(w, e), \nabla \ell(w, e))$

until convergence

- Transform w to θ

until convergence



模型部分：反例

Transition	Current State	Resulting State	Description
DROP	$[\sigma s_0, \delta, b_0 \beta, A]$	$[\sigma s_0, \delta, \beta, A]$	pops out the word that doesn't convey any semantics (e.g., function words and punctuations).
MERGE	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{b}_1 \bar{\beta}, \bar{A}]$	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	concatenates a sequence of words into a span, which can be derived as a named entity (name) or date-entity.
CONFIRM(\bar{c})	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	$[\sigma \bar{s}_0, \bar{\delta}, \bar{c} \bar{\beta}, \bar{A}]$	derives the first element of the buffer (a word or span) into a concept \bar{c} .
ENTITY(\bar{c})	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	$[\sigma \bar{s}_0, \bar{\delta}, \bar{c} \bar{\beta}, \bar{A} \cup \text{relations}(\bar{c})]$	a special form of CONFIRM that derives the first element into an entity and builds the internal entity AMR fragment.
NEW(\bar{c})	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	$[\sigma \bar{s}_0, \bar{\delta}, \bar{c} \bar{b}_0 \bar{\beta}, \bar{A}]$	generates a new concept \bar{c} and pushes it to the front of the buffer.
LEFT(\bar{r})	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A} \cup \{ \bar{s}_0 \xleftarrow{\bar{r}} \bar{b}_0 \}]$	links a relation \bar{r} between the top concepts on the stack and the buffer.
RIGHT(\bar{r})	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A} \cup \{ \bar{s}_0 \xrightarrow{\bar{r}} \bar{b}_0 \}]$	
CACHE	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	$[\sigma, \bar{s}_0 \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	passes the top concept of the stack onto the deque.
SHIFT	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	$[\sigma \bar{s}_0 \bar{\delta} \bar{b}_0, [], \bar{\beta}, \bar{A}]$	shifts the first concept of the buffer onto the stack along with those on the deque.
REDUCE	$[\sigma \bar{s}_0, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	$[\sigma, \bar{\delta}, \bar{b}_0 \bar{\beta}, \bar{A}]$	pops the top concept of the stack.

实验部分：图比表格好

LDC2014T12 Experiments

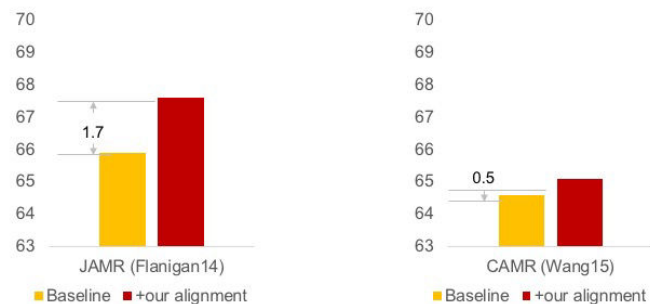
- alignment F-score

Aligner	Alignment F1 (on hand-align)	Oracle's Smatch (on dev. dataset)
JAMR	90.6	91.7
Our	95.2	94.7

- parser improvements

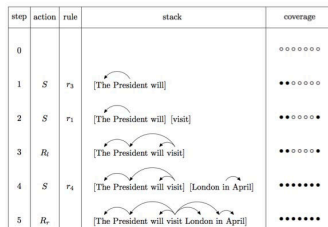
model	newswire	all
JAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	71.3	65.9
+ Our aligner	73.1	67.6
CAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	68.4	64.6
+ Our aligner	68.8	65.1

Aligner Experiments: Two Open-sourced AMR Parsers



实验部分：图比表格好

信息元素的易理解度



图

*

System	Setting	English-French	Chinese-English
GIZA++	Model 4 s2t	7.7	20.9
	Model 4 t2s	9.2	30.3
	Intersection	6.8	21.8
	Union	9.6	28.1
	Refined method	5.9	18.4
Cross-EM	HMM, joint	5.1	18.9
Vigne	Model 4 s2t	7.8	20.5
	+Model 4 t2s	5.6	18.3
	+link count	5.5	17.7
	+cross count	5.4	17.6
	+neighbor count	5.2	17.4
	+exact match	5.3	-
	+linked word count	5.2	17.3
	+bilingual dictionary	-	17.1
	+link co-occurrence count (GIZA++)	5.1	16.3
	+link co-occurrence count (Cross-EM)	4.0	15.7

表格

**

Shift-reduce parsing is efficient but suffers from parsing errors caused by syntactic ambiguity. Figure 3 shows two (partial) derivations for a dependency tree. Consider the item on the top, the algorithm can either apply a shift action to move a new item or apply a reduce left action to obtain a bigger structure. This is often referred to as **conflict** in the shift-reduce dependency parsing literature (Huang et al., 2009). In this work, the shift-reduce parser faces four types of conflicts:

正文

$$\begin{aligned}
 & \frac{\partial L(\theta)}{\partial \theta_k} \\
 &= \sum_{i=1}^I \sum_{y \in \mathcal{Y}(\mathbf{x}^{(i)})} P(y|\mathbf{x}^{(i)}; \theta) \phi_k(\mathbf{x}^{(i)}, y) \\
 & \quad - \sum_{\mathbf{x} \in \mathcal{X}} \sum_{y \in \mathcal{Y}(\mathbf{x})} P(\mathbf{x}, y; \theta) \phi_k(\mathbf{x}, y) \\
 &= \sum_{i=1}^I \mathbb{E}_{y|\mathbf{x}^{(i)}; \theta} [\phi_k(\mathbf{x}^{(i)}, y)] - \mathbb{E}_{\mathbf{x}, y; \theta} [\phi_k(\mathbf{x}, y)]
 \end{aligned}$$

公式

```

Algorithm 1 A beam search algorithm for word alignment
1: procedure ALIGN(l, e)
2:   open ← ∅                                ▷ a list of active alignments
3:   N ← ∅                                    ▷ n-best list
4:   a ← ∅                                    ▷ begin with an empty alignment
5:   ADD(open, a, ∅, ∅)                       ▷ initialize the list
6:   while open ≠ ∅ do
7:     closed ← ∅                             ▷ a list of promising alignments
8:     for all a ∈ open do
9:       for all l' ∈ l × l - a do            ▷ enumerate all possible new links
10:        a' ← a ∪ {l'}                      ▷ produce a new alignment
11:        g ← GAIN(l, e, a, l')              ▷ compute the link gain
12:        if g > 0 then                       ▷ ensure that the score will increase
13:          ADD(closed, a', ∅, ∅)             ▷ update promising alignments
14:        end if
15:        ADD(N, a', 0, n)                   ▷ update n-best list
16:      end for
17:    end for
18:    open ← closed                           ▷ update active alignments
19:  end while
20:  return N                                  ▷ return n-best list
21: end procedure
  
```

算法

Proof of Theorem 1: Let $\bar{\alpha}^k$ be the weights before the k 'th mistake is made. It follows that $\bar{\alpha}^1 = 0$. Suppose the k 'th mistake is made at the i 'th example. Take z to the output proposed at this example, $z = \arg\max_{y \in \text{GEN}(x_i)} \Phi(x_i, y) \cdot \bar{\alpha}^k$. It follows from the algorithm updates that $\bar{\alpha}^{k+1} = \bar{\alpha}^k + \Phi(x_i, y_i) - \Phi(x_i, z)$. We take inner products of both sides with the vector \mathbf{U} :


$$\begin{aligned}
 \mathbf{U} \cdot \bar{\alpha}^{k+1} &= \mathbf{U} \cdot \bar{\alpha}^k + \mathbf{U} \cdot \Phi(x_i, y_i) - \mathbf{U} \cdot \Phi(x_i, z) \\
 &\geq \mathbf{U} \cdot \bar{\alpha}^k + \delta
 \end{aligned}$$

where the inequality follows because of the property of \mathbf{U} assumed in Eq. 3. Because $\bar{\alpha}^1 = 0$, and therefore $\mathbf{U} \cdot \bar{\alpha}^1 = 0$, it follows by induction on k that for all k , $\mathbf{U} \cdot \bar{\alpha}^{k+1} \geq k\delta$. Because $\mathbf{U} \cdot \bar{\alpha}^{k+1} \leq \|\mathbf{U}\| \|\bar{\alpha}^{k+1}\|$, it follows that $\|\bar{\alpha}^{k+1}\| \geq k\delta$.

证明

实验部分：图比表格好

信息元素的易理解度



图

Item	Setting	Single Input	Single Output
Item 1	Item 1	1.0	1.0
Item 2	Item 2	1.0	1.0
Item 3	Item 3	1.0	1.0
Item 4	Item 4	1.0	1.0
Item 5	Item 5	1.0	1.0
Item 6	Item 6	1.0	1.0
Item 7	Item 7	1.0	1.0
Item 8	Item 8	1.0	1.0
Item 9	Item 9	1.0	1.0
Item 10	Item 10	1.0	1.0
Item 11	Item 11	1.0	1.0
Item 12	Item 12	1.0	1.0
Item 13	Item 13	1.0	1.0
Item 14	Item 14	1.0	1.0
Item 15	Item 15	1.0	1.0
Item 16	Item 16	1.0	1.0
Item 17	Item 17	1.0	1.0
Item 18	Item 18	1.0	1.0
Item 19	Item 19	1.0	1.0
Item 20	Item 20	1.0	1.0
Item 21	Item 21	1.0	1.0
Item 22	Item 22	1.0	1.0
Item 23	Item 23	1.0	1.0
Item 24	Item 24	1.0	1.0
Item 25	Item 25	1.0	1.0
Item 26	Item 26	1.0	1.0
Item 27	Item 27	1.0	1.0
Item 28	Item 28	1.0	1.0
Item 29	Item 29	1.0	1.0
Item 30	Item 30	1.0	1.0
Item 31	Item 31	1.0	1.0
Item 32	Item 32	1.0	1.0
Item 33	Item 33	1.0	1.0
Item 34	Item 34	1.0	1.0
Item 35	Item 35	1.0	1.0
Item 36	Item 36	1.0	1.0
Item 37	Item 37	1.0	1.0
Item 38	Item 38	1.0	1.0
Item 39	Item 39	1.0	1.0
Item 40	Item 40	1.0	1.0
Item 41	Item 41	1.0	1.0
Item 42	Item 42	1.0	1.0
Item 43	Item 43	1.0	1.0
Item 44	Item 44	1.0	1.0
Item 45	Item 45	1.0	1.0
Item 46	Item 46	1.0	1.0
Item 47	Item 47	1.0	1.0
Item 48	Item 48	1.0	1.0
Item 49	Item 49	1.0	1.0
Item 50	Item 50	1.0	1.0
Item 51	Item 51	1.0	1.0
Item 52	Item 52	1.0	1.0
Item 53	Item 53	1.0	1.0
Item 54	Item 54	1.0	1.0
Item 55	Item 55	1.0	1.0
Item 56	Item 56	1.0	1.0
Item 57	Item 57	1.0	1.0
Item 58	Item 58	1.0	1.0
Item 59	Item 59	1.0	1.0
Item 60	Item 60	1.0	1.0
Item 61	Item 61	1.0	1.0
Item 62	Item 62	1.0	1.0
Item 63	Item 63	1.0	1.0
Item 64	Item 64	1.0	1.0
Item 65	Item 65	1.0	1.0
Item 66	Item 66	1.0	1.0
Item 67	Item 67	1.0	1.0
Item 68	Item 68	1.0	1.0
Item 69	Item 69	1.0	1.0
Item 70	Item 70	1.0	1.0
Item 71	Item 71	1.0	1.0
Item 72	Item 72	1.0	1.0
Item 73	Item 73	1.0	1.0
Item 74	Item 74	1.0	1.0
Item 75	Item 75	1.0	1.0
Item 76	Item 76	1.0	1.0
Item 77	Item 77	1.0	1.0
Item 78	Item 78	1.0	1.0
Item 79	Item 79	1.0	1.0
Item 80	Item 80	1.0	1.0
Item 81	Item 81	1.0	1.0
Item 82	Item 82	1.0	1.0
Item 83	Item 83	1.0	1.0
Item 84	Item 84	1.0	1.0
Item 85	Item 85	1.0	1.0
Item 86	Item 86	1.0	1.0
Item 87	Item 87	1.0	1.0
Item 88	Item 88	1.0	1.0
Item 89	Item 89	1.0	1.0
Item 90	Item 90	1.0	1.0
Item 91	Item 91	1.0	1.0
Item 92	Item 92	1.0	1.0
Item 93	Item 93	1.0	1.0
Item 94	Item 94	1.0	1.0
Item 95	Item 95	1.0	1.0
Item 96	Item 96	1.0	1.0
Item 97	Item 97	1.0	1.0
Item 98	Item 98	1.0	1.0
Item 99	Item 99	1.0	1.0
Item 100	Item 100	1.0	1.0

表格

Shift-reduce parsing is efficient but suffers from parsing errors caused by syntactic ambiguity. Figure 7 shows two (partial) derivations for a dependency tree. Consider the item on the top; the algorithm can either apply a shift action to move a new item or apply a reduce-left action to obtain a bigger structure. This is often referred to as a *conflict* in the shift-reduce dependency parsing literature (Huang et al., 2009). In this work, the shift-reduce parser faces four types of conflicts:

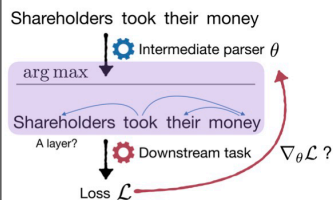
正文

用图与例子来描述方法和实验

结论部分：新的展现形式

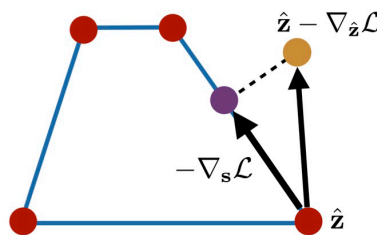
Conclusion

Problem

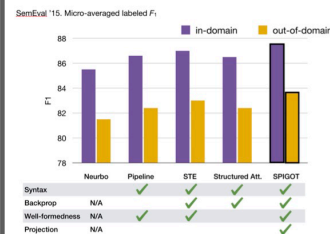


Method

SPIGOT

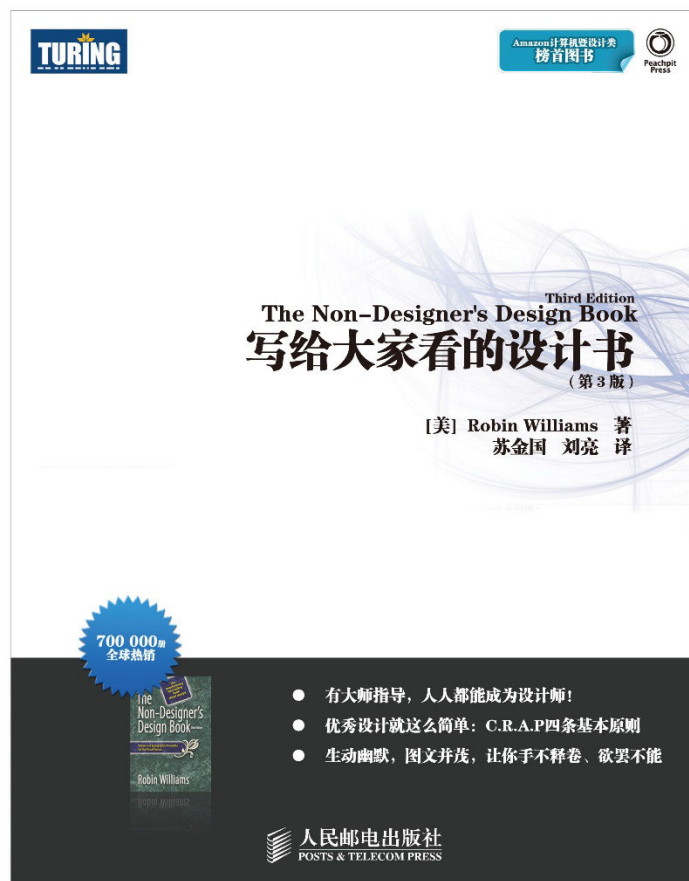


Results



设计原则

- 亲密性：相关的元素应该组织到一起
- 重复：相同的内容达到形式的统一
- 对比：如果两项不完全相同，就应使之截然不同
- 对齐：使元素之间产生关联，有关联的都应对齐



根据设计原则做幻灯片

Challenges and Contribution

- The first challenge is deriving an optimal alignment in ambiguous situations.
- The second challenge is recalling more semantically matched word-concept pair without harming the alignment precision.
- The final challenge which is faced by both the rule-based and unsupervised aligners is tuning the alignment with downstream parser learning.
- We proposed an enhanced aligner tuned by transition-based oracle parser

加入空行提高相关
元素的亲密性

Challenges and Contribution

- The first challenge is deriving an optimal alignment in ambiguous situations.
- The second challenge is recalling more semantically matched word-concept pair without harming the alignment precision.
- The final challenge which is faced by both the rule-based and unsupervised aligners is tuning the alignment with downstream parser learning.
- We proposed an enhanced aligner tuned by transition-based oracle parser

Challenges and Contribution

- **Challenges**
 - deriving an optimal alignment in ambiguous situations.
 - recalling more semantically matched word-concept pair without harming the alignment precision.
 - tuning the alignment with downstream parser learning.
- **Contribution**
 - an enhanced aligner tuned by transition-based oracle parser

相同内容使用相同样式
即提高了一致性又形成
了必要的对比

避免不对齐

Our aligner algorithm

- Enhancing aligner with rich semantic resources
- Producing multiple alignments

Input: An AMR graph with a set of graph fragments C ; a sentence W ; a set of matching rules P_M ; and a set of updating rules P_U .

Output: a set of alignments \mathcal{A} .

```
1 for  $c \in C$  do
2    $A_c \leftarrow \emptyset$ ;
3 for  $\rho_M \in P_M$  do
4   for  $w_{s,e} \leftarrow \text{spans}(W)$  do
5     for  $c \in C$  do
6       if  $\rho_M(c, w_{s,e})$  then
7          $A_c \leftarrow A_c \cup (s, e, \text{nil})$ ;
8 updated  $\leftarrow \text{true}$ ;
9 while updated is true do
10   updated  $\leftarrow \text{false}$ ;
11   for  $\rho_U \in P_U$  do
12     for  $c, c' \in C \times C$  do
13       for  $(s, e, d) \in A_c^t$  do
14         if  $\rho_U(c, w_{s,e}) \wedge (s, e, c') \notin A_{c'}$  then
15            $A_{c'} \leftarrow A_{c'} \cup (s, e, c')$ ;
16           updated  $\leftarrow \text{true}$ ;
17  $\mathcal{A} \leftarrow \emptyset$ ;
18 for  $(a_1, \dots, a_c) \in \text{CartesianProduct}(A_1, \dots, A_{|C|})$  do
19   legal  $\leftarrow \text{true}$ ;
20   for  $a \in (a_1, \dots, a_c)$  do
21      $(s, e, c') \leftarrow a$ ;
22      $(s', e', d) \leftarrow a_{c'}$ ;
23     if  $s \neq s' \wedge e \neq e'$  then
24       legal  $\leftarrow \text{false}$ ;
25   if legal then
26      $\mathcal{A} \leftarrow \mathcal{A} \cup (a_1, \dots, a_c)$ ;
```

“乱” 的原因：视线跳动过多

Experiments

- We conduct experiments on LDC2014T12
- We evaluate the alignment F-score and Smatch of resulted parsers

Aligner	Alignment F1 (on hand-align)	Oracle's Smatch (on dev. dataset)
JAMR	90.6	91.7
Our	95.2	94.7

model	newswire	all
JAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	71.3	65.9
+ Our aligner	73.1	67.6
CAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	68.4	64.6
+ Our aligner	68.8	65.1

model	newswire	all
Our single parser: Word only		
+ JAMR aligner	68.6	63.9
+ Our aligner	69.3	64.7
Our single parser: Word, POS		
+ JAMR aligner	68.8	64.6
+ Our aligner	69.8	65.2
Our ensemble: Word only + Our aligner		
x3	71.9	67.4
x10	72.5	68.1
Our ensemble: Word, POS + Our aligner		
x3	72.5	67.7
x10	73.3	68.4

“乱” 的原因：视线跳动过多

Experiments

- We conduct experiments on LDC2014T12
- We evaluate the alignment F-score and Smatch of resulted parsers

Aligner	Alignment F1 (on hand-align)	Oracle's Smatch (on dev. dataset)
JAMR	90.6	91.7
Our	95.2	94.7

model	newswire	all
JAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	71.3	65.9
+ Our aligner	73.1	67.6
CAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	68.4	64.6
+ Our aligner	68.8	65.1

model	newswire	all
Our single parser: Word only		
+ JAMR aligner	68.6	63.9
+ Our aligner	69.3	64.7
Our single parser: Word, POS		
+ JAMR aligner	68.8	64.6
+ Our aligner	69.8	65.2
Our ensemble: Word only + Our aligner		
x3	71.9	67.4
x10	72.5	68.1
Our ensemble: Word, POS + Our aligner		
x3	72.5	67.7
x10	73.3	68.4

“乱” 的解法：重新组织内容

Experiments

- We conduct experiments on LDC2014T12
- We evaluate the alignment F-score and Smatch of resulted parsers

Aligner	Alignment F1 (on hand-align)	Oracle's Smatch (on dev. dataset)
JAMR	90.6	91.7
Our	95.2	94.7

model	newswire	all
JAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	71.3	65.9
+ Our aligner	73.1	67.6
CAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	68.4	64.6
+ Our aligner	68.8	65.1

model	newswire	all
Our single parser: Word only		
+ JAMR aligner	68.6	63.9
+ Our aligner	69.3	64.7
Our single parser: Word, POS		
+ JAMR aligner	68.8	64.6
+ Our aligner	69.8	65.2
Our ensemble: Word only + Our aligner		
x3	71.9	67.4
x10	72.5	68.1
Our ensemble: Word, POS + Our aligner		
x3	72.5	67.7
x10	73.3	68.4

LDC2014T12 Experiments

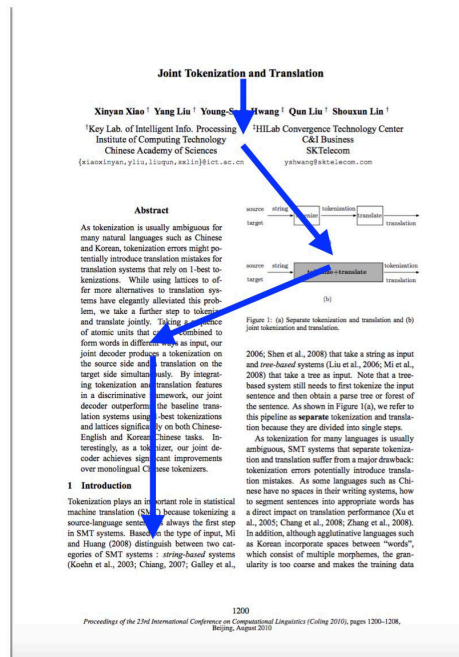
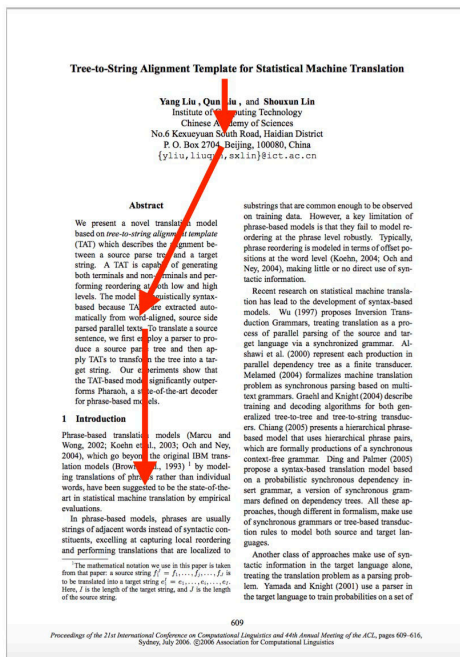
- alignment F-score
- parser improvements

Aligner	Alignment F1 (on hand-align)	Oracle's Smatch (on dev. dataset)
JAMR	90.6	91.7
Our	95.2	94.7

model	newswire	all
JAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	71.3	65.9
+ Our aligner	73.1	67.6
CAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	68.4	64.6
+ Our aligner	68.8	65.1

视线跳动在论文写作中的作用

信息流的变化




参考文献

- Simon Peyton Jones: How to give a great talk
- 写给大家看的设计书
- 机器翻译学术论文写作方法与技巧
- 知乎专栏：跟我学个P

总结

(Zhang and Nivre 2011, Martins et al 2013)

 **Our Work**

- A neural network based dependency parser!


Parsing on English Penn Treebank (§23):

		Unlabeled attachment score (UAS)	sent / s
Transition -based	MaltParser (greedy)	89.9	560
	Our Parser (greedy)	92.0	1013
	Zpar: beam = 64	92.9*	29*
Graph -based	MSTParser	92.0	12
	TurboParser	93.1*	31*

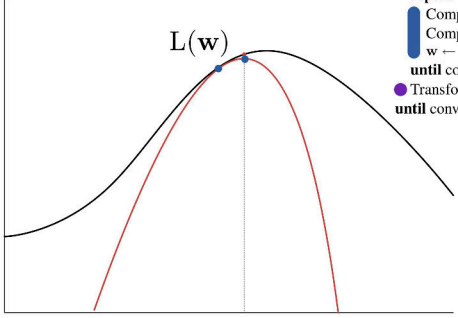
A Fast and Accurate Dependency Parser using Neural Networks

3

为了抓住听众，把最好的部分前置

 **EM with Features**

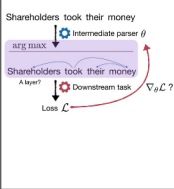
Initialize weights w
repeat
 • Compute expected counts e
repeat
 • Compute $\ell(w, e)$
 • Compute $\nabla \ell(w, e)$
 $w \leftarrow \text{climb}(w, \ell(w, e), \nabla \ell(w, e))$
until convergence
 • Transform w to θ
until convergence



模型部分有取舍，用好图和例子

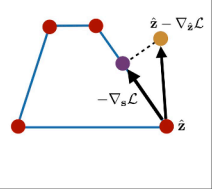
Conclusion

Problem

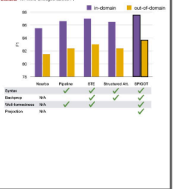


Method

SPIGOT



Results



“结论”也有新思路

Challenges and Contributions

- Challenges**
 - deriving an optimal alignment in ambiguous sentences
 - recalling more semantically matched word-concept pair without harming the alignment precision.
 - tuning the alignment with downstream parser
- Contribution**
 - an enhanced aligner tuned by transition-based

亲密性

重复

对比

对齐

四项设计的基本原则

祝大家产出优秀的学术工作