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

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。

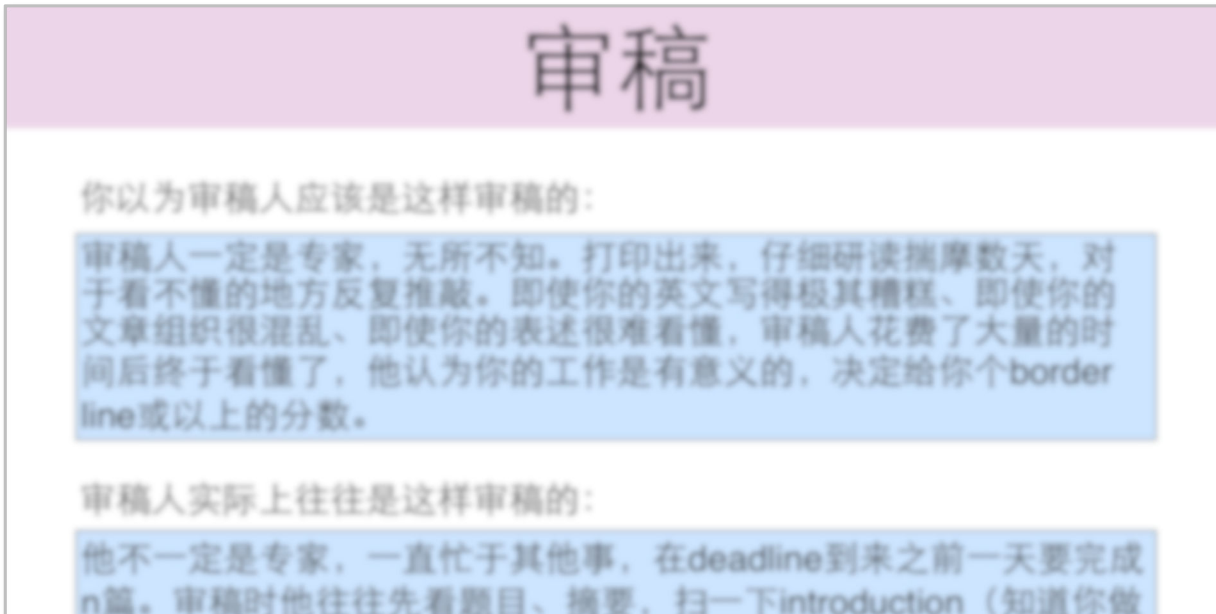
审稿人实际上往往是这样审稿的:

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

刘洋. 2014. 机器翻译学术论文写作方法与技巧

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 Nirve 2011, Martins et al 2013)

Our Work

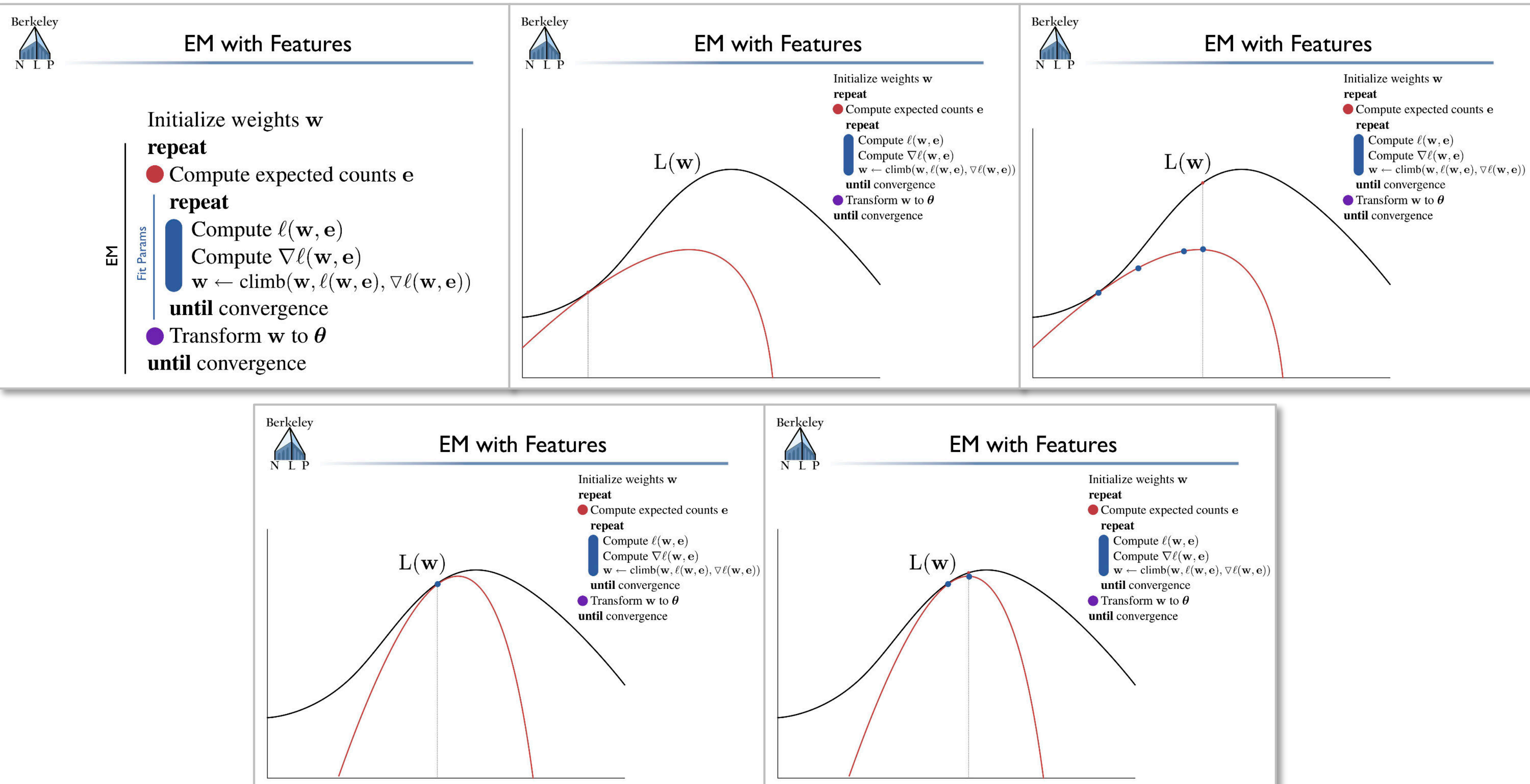
- A neural network based dependency parser!

Parsing on English Penn Treebank (§23):

		Unlabeled attachment score (UAS)	sent / s
Transition -based	MaltParser (greedy)	89.9 (+2.1)	560 (×1.8)
	Our Parser (greedy)	92.0	1013
	Zpar: beam = 64	92.9*	29*
Graph -based	MSTParser	92.0	12
	TurboParser	93.1*	31*

Danqi Chen and Christopher Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks，第三页

模型部分：多用例子



Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless Unsupervised Learning with Features，第28到54页

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	$[\sigma\|s_0,\ \delta,\ b_0\|b_1\|\beta,\ A]$	$[\sigma\|s_0,\ \delta,\ b_0\_b_1\|\beta,\ A]$	concatenates a sequence of words into a span, which can be derived as a named entity (`name`) or `date-entity`.
CONFIRM(c)	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A]$	$[\sigma\|s_0,\ \delta,\ c\|\beta,\ A]$	derives the first element of the buffer (a word or span) into a concept `c`.
ENTITY(c)	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A]$	$[\sigma\|s_0,\ \delta,\ c\|\beta,\ A \cup \text{relations}(c)]$	a special form of CONFIRM that derives the first element into an entity and builds the internal entity AMR fragment.
NEW(c)	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A]$	$[\sigma\|s_0,\ \delta,\ c\|b_0\|\beta,\ A]$	generates a new concept `c` and pushes it to the front of the buffer.
LEFT(r)	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A]$	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A \cup \{s_0 \xleftarrow{r} b_0\}]$	links a relation `r` between the top concepts on the stack and the buffer.
RIGHT(r)	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A]$	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A \cup \{s_0 \xrightarrow{r} b_0\}]$	
CACHE	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A]$	$[\sigma,\ s_0\|\delta,\ b_0\|\beta,\ A]$	passes the top concept of the stack onto the deque.
SHIFT	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A]$	$[\sigma\|s_0\|\delta\|b_0,\ [\,],\ \beta,\ A]$	shifts the first concept of the buffer onto the stack along with those on the deque.
REDUCE	$[\sigma\|s_0,\ \delta,\ b_0\|\beta,\ A]$	$[\sigma,\ \delta,\ b_0\|\beta,\ A]$	pops the top concept of the stack.

实验部分：图比表格好

LDC2014T12 Experiments

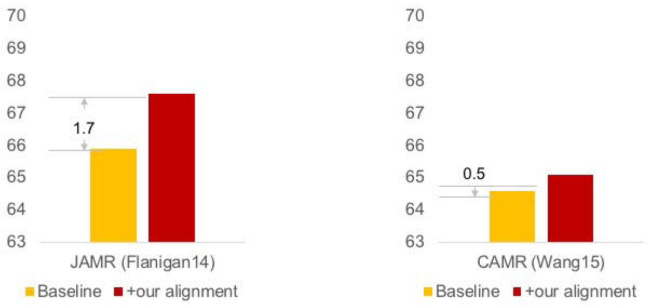
- alignment F-score

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

- parser improvements

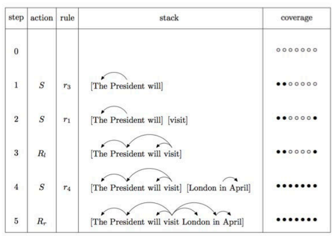
model	newswire	all
JAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	71.3	65.9
+ Our aligner	73.1	67.6
CAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	68.4	64.6
+ Our aligner	68.8	65.1

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

***

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

公式

****

Algorithm 1 A beam search algorithm for word alignment

```
procedure ALIGN(f, e)
    open ← ∅                                   ▷ a list of active alignments
    N ← ∅                                      ▷ n-best list
    a ← ∅                                      ▷ begin with an empty alignment
    ADD(open, a, β, b)                         ▷ initialize the list
    while open ≠ ∅ do
        closed ← ∅                             ▷ a list of promising alignments
        for all a ∈ open do
            for all l ∈ J × I − a do           ▷ enumerate all possible new links
                a′ ← a ∪ {l}                   ▷ produce a new alignment
                g ← GAIN(f, e, a, l)           ▷ compute the link gain
                if g > 0 then                  ▷ ensure that the score will increase
                    ADD(closed, a′, β, b)      ▷ update promising alignments
                end if
                ADD(N, a′, 0, n)               ▷ update n-best list
            end for
        end for
        open ← closed                          ▷ update active alignments
    end while
    return N                                   ▷ return n-best list
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, $z = \arg\max_{y \in \mathbf{GEN}(x_i)} \Phi(x_i, y) \cdot \bar{\alpha}^k$. 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 $\mathbf{U}$:

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

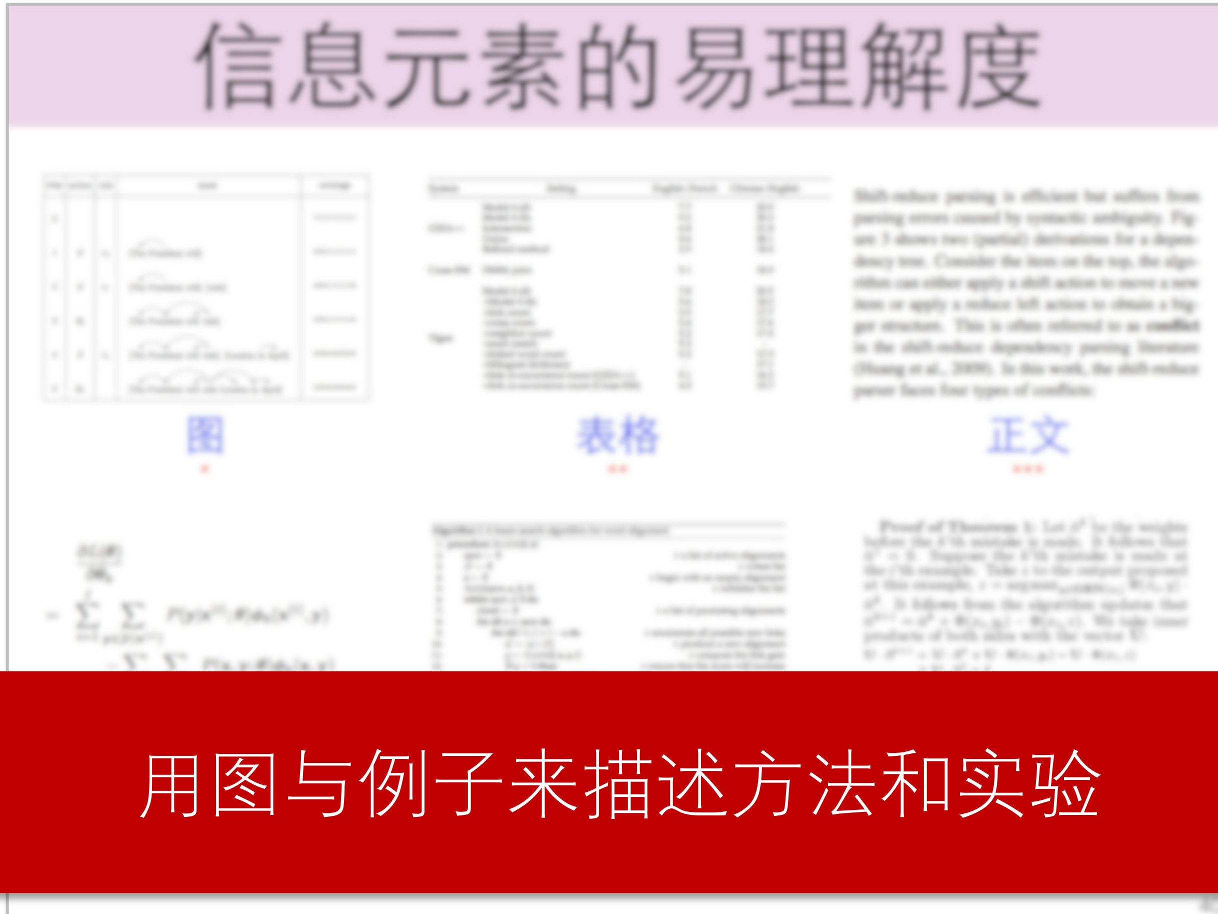
where the inequality follows because of the property of $\mathbf{U}$ assumed in Eq. 3. Because $\bar{\alpha}^1 = 0$, and therefore $\mathbf{U} \cdot \bar{\alpha}^1 = 0$, it follows by induction on k that for all k, $\mathbf{U} \cdot \bar{\alpha}^{k+1} \geq k\delta$. Because $\mathbf{U} \cdot \bar{\alpha}^{k+1}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

证明

******

刘洋. 2014. 机器翻译学术论文写作方法与技巧

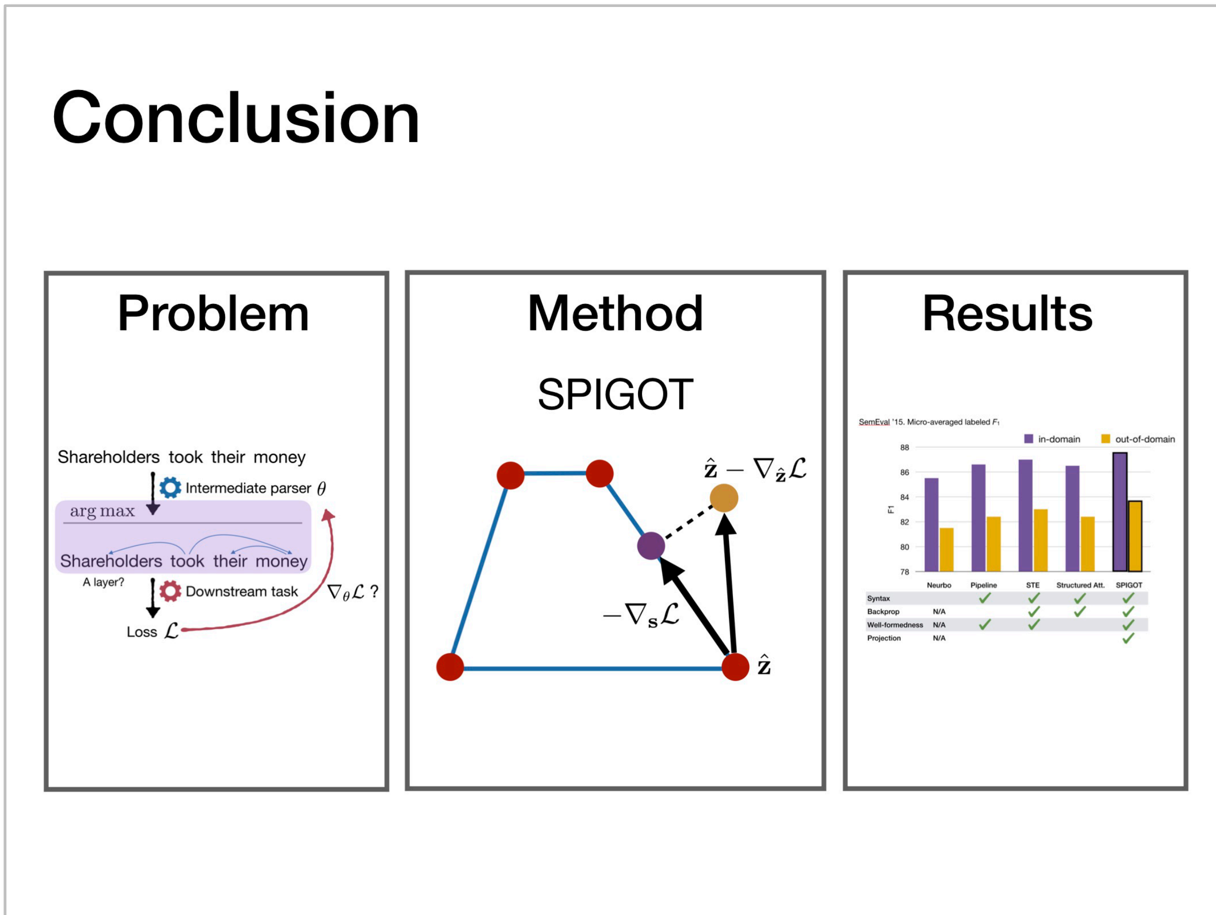
实验部分：图比表格好



用图与例子来描述方法和实验

刘洋. 2014. 机器翻译学术论文写作方法与技巧

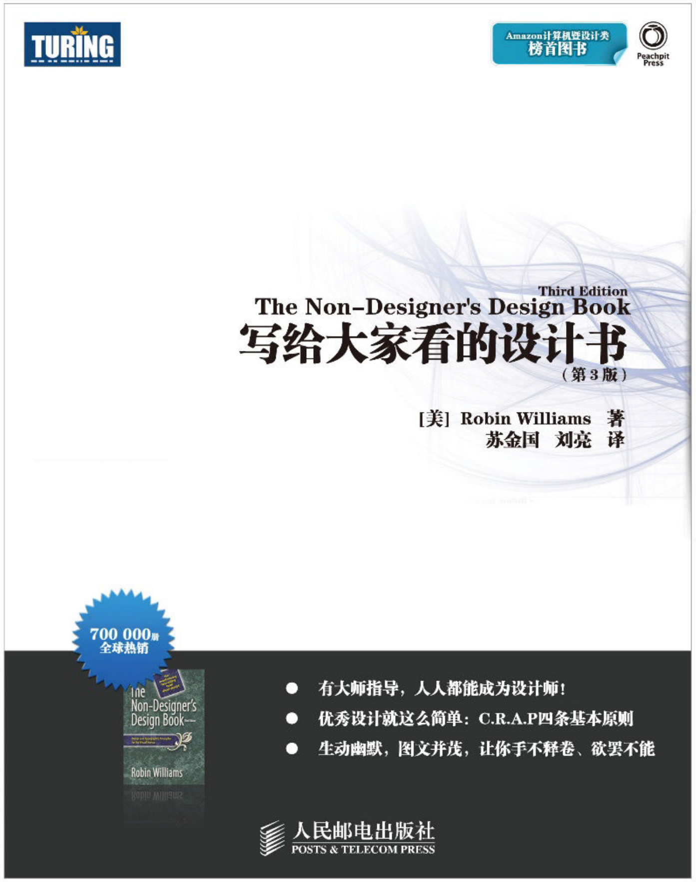
结论部分：新的展现形式



Hao Peng, Sam Thomson, and Noah A. Smith. 2018. Backpropagating through Structured Argmax using a SPIGOT，最后一页

设计原则

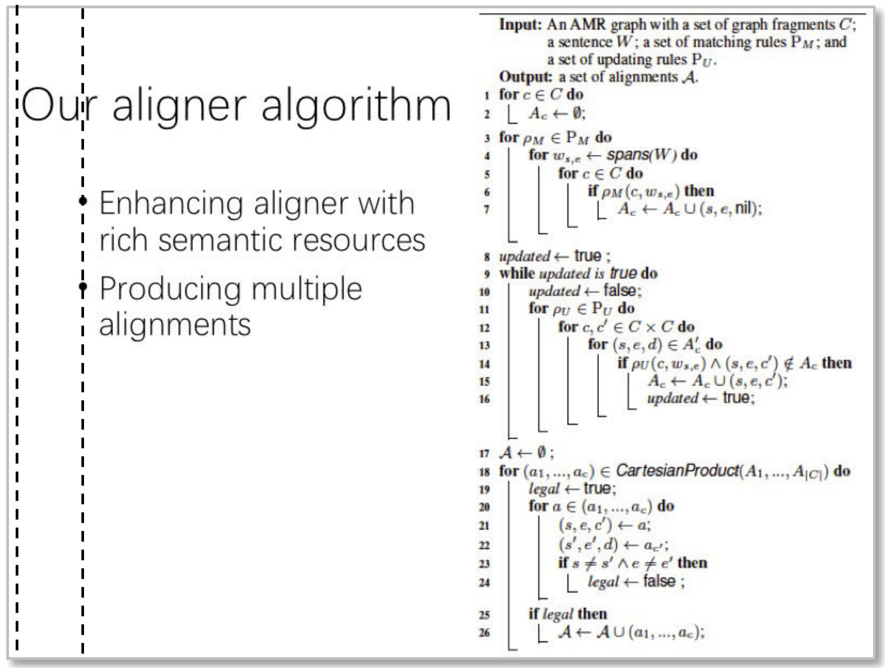
- 亲密性：相关的元素应该组织到一起
- 重复：相同的内容达到形式的统一
- 对比：如果两项不完全相同，就应使之截然不同
- 对齐：使元素之间产生关联，有关联的都应对齐



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
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
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

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 𝒜.
for c ∈ C do
    A_c ← ∅;
for ρ_M ∈ P_M do
    for w_{s,e} ← spans(W) do
        for c ∈ C do
            if ρ_M(c, w_{s,e}) then
                A_c ← A_c ∪ (s, e, nil);
updated ← true ;
while updated is true do
    updated ← false;
    for ρ_U ∈ P_U do
        for c, c′ ∈ C × C do
            for (s, e, d) ∈ A′_c do
                if ρ_U(c, w_{s,e}) ∧ (s, e, c′) ∉ A_c then
                    A_c ← A_c ∪ (s, e, c′);
                    updated ← true;
𝒜 ← ∅ ;
for (a_1, ..., a_c) ∈ CartesianProduct(A_1, ..., A_|C|) do
    legal ← true;
    for a ∈ (a_1, ..., a_c) do
        (s, e, c′) ← a;
        (s′, e′, d) ← a_c′;
        if s ≠ s′ ∧ e ≠ e′ then
            legal ← false ;
    if legal then
        𝒜 ← 𝒜 ∪ (a_1, ..., 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

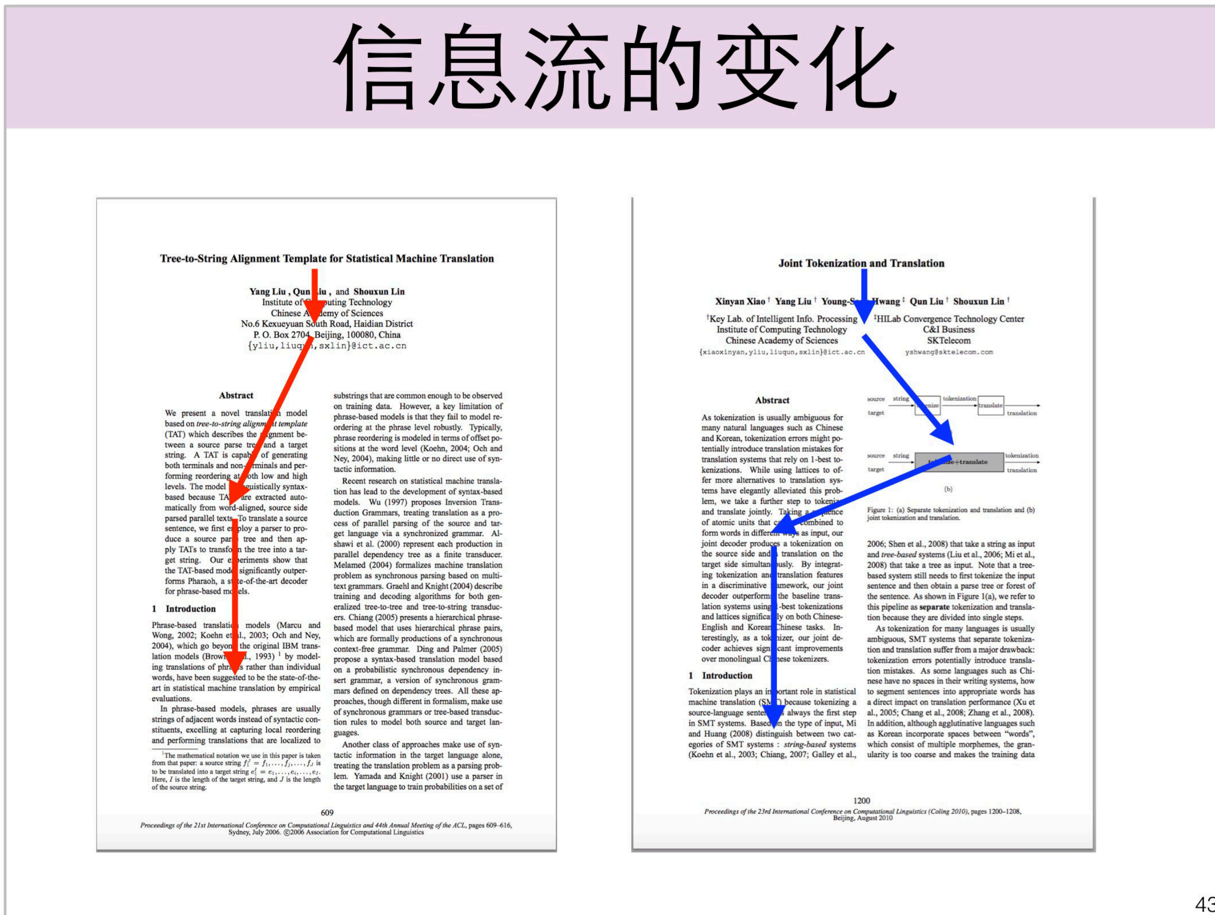
- alignment F-score

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

视线跳动在论文写作中的作用

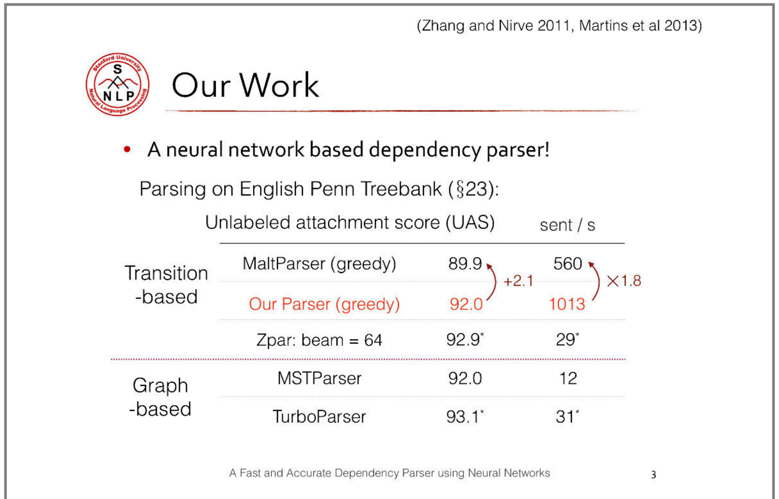


刘洋. 2014. 机器翻译学术论文写作方法与技巧

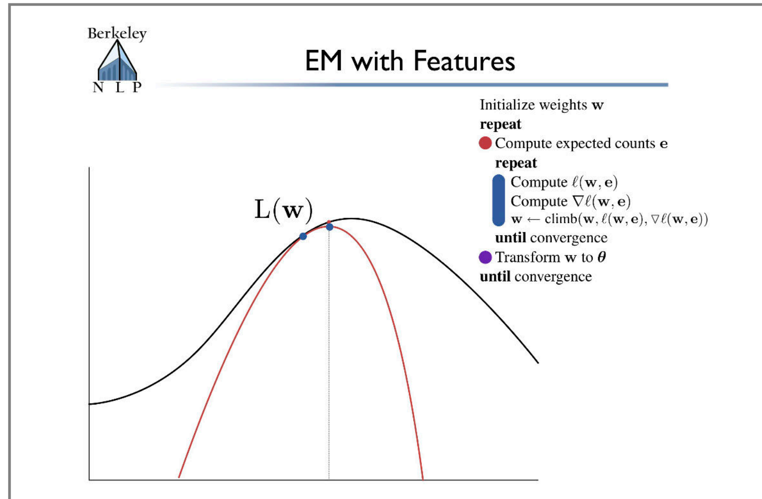
参考文献

- 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

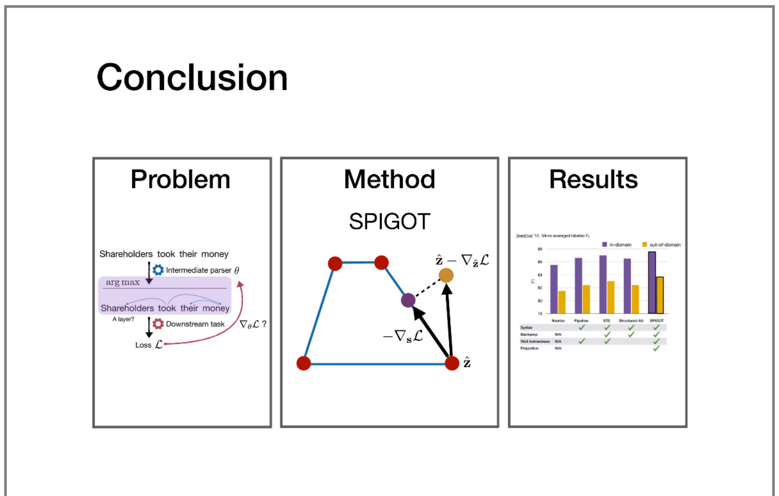
总结



为了抓住听众，把最好的部分前置



模型部分有取舍，用好图和例子



“结论”也有新思路

四项设计的基本原则

祝大家产出优秀的学术工作