论文写作的易读性原则

案例分析：基于Seq2Seq的对话数据增广

报告人：刘一佳
合作者：侯宇泰、车万翔、刘挺

http://yjliu.net/cv/res/2018-08-19-nlpcc-sws.compressed.pdf

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

简介

模型

模型

模型

实验

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borderline或以上的分数。

审稿人实际上往往是这样审稿的：

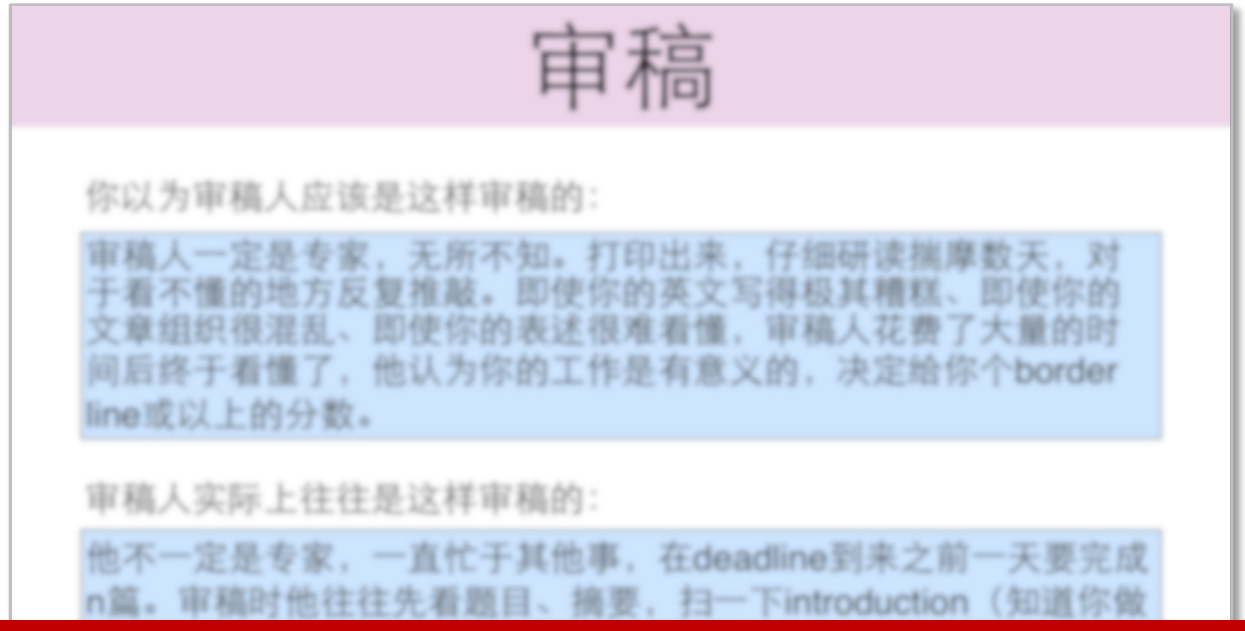
他不一定是专家，一直忙于其他事，在deadline到来之前一天要完成n篇。审稿时他往往先看题目、摘要，扫一下introduction（知道你做什么），然后直接翻到最后找核心实验结果（做得好不好），然后基本确定录还是不录（也许只用5分钟！）。如果决定录，剩下就是写些赞美的话，指出些次要的小毛病。如果决定拒，下面的过程就是细看中间部分找理由拒了。

第一印象定录拒，5分钟内打动审稿人

12

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*

刘洋. 2014. 机器翻译学术论文写作方法与技巧

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!

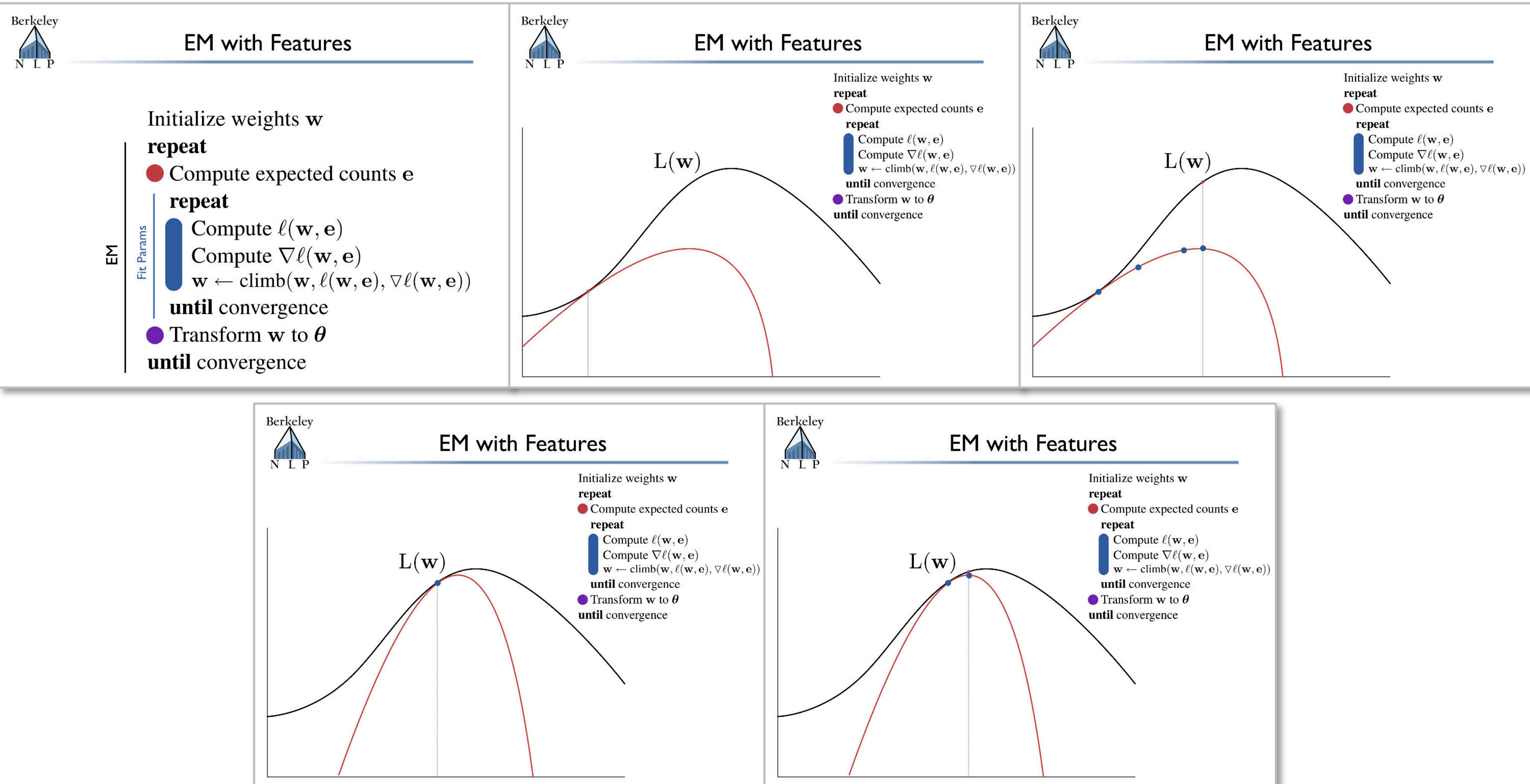
Parsing on English Penn Treebank (§23):

		Unlabeled attachment score (UAS)	sent / s
Transition -based	MaltParser (greedy)	89.9	560
		+2.1	×1.8
	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

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	$[\sigma \mid \mathtt{s}_0,\ \delta,\ b_0 \mid \beta,\ A]$	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \beta,\ A]$	pops out the word that doesn't convey any semantics (e.g., function words and punctuations).
MERGE	$[\sigma \mid \mathtt{s}_0,\ \delta,\ b_0 \mid b_1 \mid \beta,\ A]$	$[\sigma \mid \mathtt{s}_0,\ \delta,\ b_0\_b_1 \mid \beta,\ A]$	concatenates a sequence of words into a span, which can be derived as a named entity (`name`) or `date-entity`.
CONFIRM(c)	$[\sigma \mid \mathtt{s}_0,\ \delta,\ b_0 \mid \beta,\ A]$	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{c} \mid \beta,\ A]$	derives the first element of the buffer (a word or span) into a concept `c`.
ENTITY(c)	$[\sigma \mid \mathtt{s}_0,\ \delta,\ b_0 \mid \beta,\ A]$	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{c} \mid \beta,\ A \cup \text{relations}(c)]$	a special form of CONFIRM that derives the first element into an entity and builds the internal entity AMR fragment.
NEW(c)	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{b}_0 \mid \beta,\ A]$	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{c} \mid \mathtt{b}_0 \mid \beta,\ A]$	generates a new concept `c` and pushes it to the front of the buffer.
LEFT(r)	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{b}_0 \mid \beta,\ A]$	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{b}_0 \mid \beta,\ A \cup \{\mathtt{s}_0 \xleftarrow{\mathtt{r}} \mathtt{b}_0\}]$	links a relation `r` between the top concepts on the stack and the buffer.
RIGHT(r)	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{b}_0 \mid \beta,\ A]$	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{b}_0 \mid \beta,\ A \cup \{\mathtt{s}_0 \xrightarrow{\mathtt{r}} \mathtt{b}_0\}]$	
CACHE	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{b}_0 \mid \beta,\ A]$	$[\sigma,\ \mathtt{s}_0 \mid \delta,\ \mathtt{b}_0 \mid \beta,\ A]$	passes the top concept of the stack onto the deque.
SHIFT	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{b}_0 \mid \beta,\ A]$	$[\sigma \mid \mathtt{s}_0 \mid \delta \mid \mathtt{b}_0,\ [\,],\ \beta,\ A]$	shifts the first concept of the buffer onto the stack along with those on the deque.
REDUCE	$[\sigma \mid \mathtt{s}_0,\ \delta,\ \mathtt{b}_0 \mid \beta,\ A]$	$[\sigma,\ \delta,\ \mathtt{b}_0 \mid \beta,\ A]$	pops the top concept of the stack.

实验部分：图比表格好

LDC2014T12 Experiments

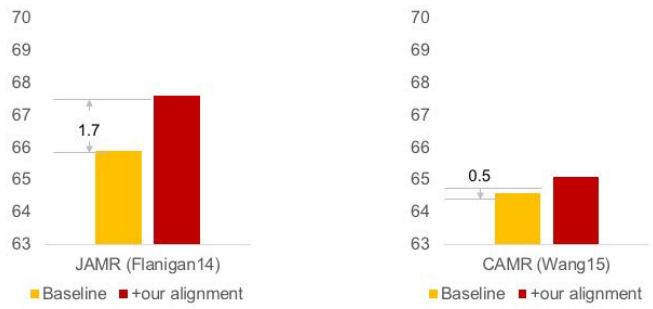
- alignment F-score

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

- parser improvements

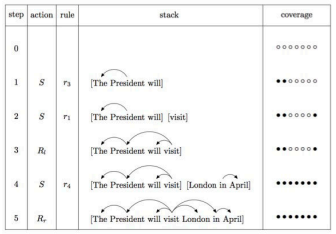
model	newswire	all
JAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	71.3	65.9
+ Our aligner	73.1	67.6
CAMR parser: Word, POS, NER, DEP		
+ JAMR aligner	68.4	64.6
+ Our aligner	68.8	65.1

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

***

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} = \sum_{i=1}^{I} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}^{(i)})} P(\mathbf{y}|\mathbf{x}^{(i)};\boldsymbol{\theta})\boldsymbol{\phi}_k(\mathbf{x}^{(i)},\mathbf{y}) - \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} P(\mathbf{x},\mathbf{y};\boldsymbol{\theta})\boldsymbol{\phi}_k(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{I} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(i)};\boldsymbol{\theta}}[\boldsymbol{\phi}_k(\mathbf{x}^{(i)},\mathbf{y})] - \mathbb{E}_{\mathbf{x},\mathbf{y};\boldsymbol{\theta}}[\boldsymbol{\phi}_k(\mathbf{x},\mathbf{y})]$$

公式

****

Algorithm 1 A beam search algorithm for word alignment

```
1: procedure ALIGN(f, e)
2:     open ← ∅                                   ▷ a list of active alignments
3:     N ← ∅                                      ▷ n-best list
4:     a ← ∅                                      ▷ begin with an empty alignment
5:     ADD(open, a, β, b)                         ▷ initialize the list
6:     while open ≠ ∅ do
7:         closed ← ∅                             ▷ a list of promising alignments
8:         for all a ∈ open do
9:             for all l ∈ J × I − a do           ▷ enumerate all possible new links
10:                a' ← a ∪ {l}                   ▷ produce a new alignment
11:                g ← GAIN(f, e, a, l)           ▷ compute the link gain
12:                if g > 0 then                  ▷ ensure that the score will increase
13:                    ADD(closed, a', β, b)      ▷ 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, $z = \text{argmax}_{y \in \mathbf{GEN}(x_i)} \Phi(x_i, y) \cdot \bar{\alpha}^k$. 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 $\mathbf{U}$:

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

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

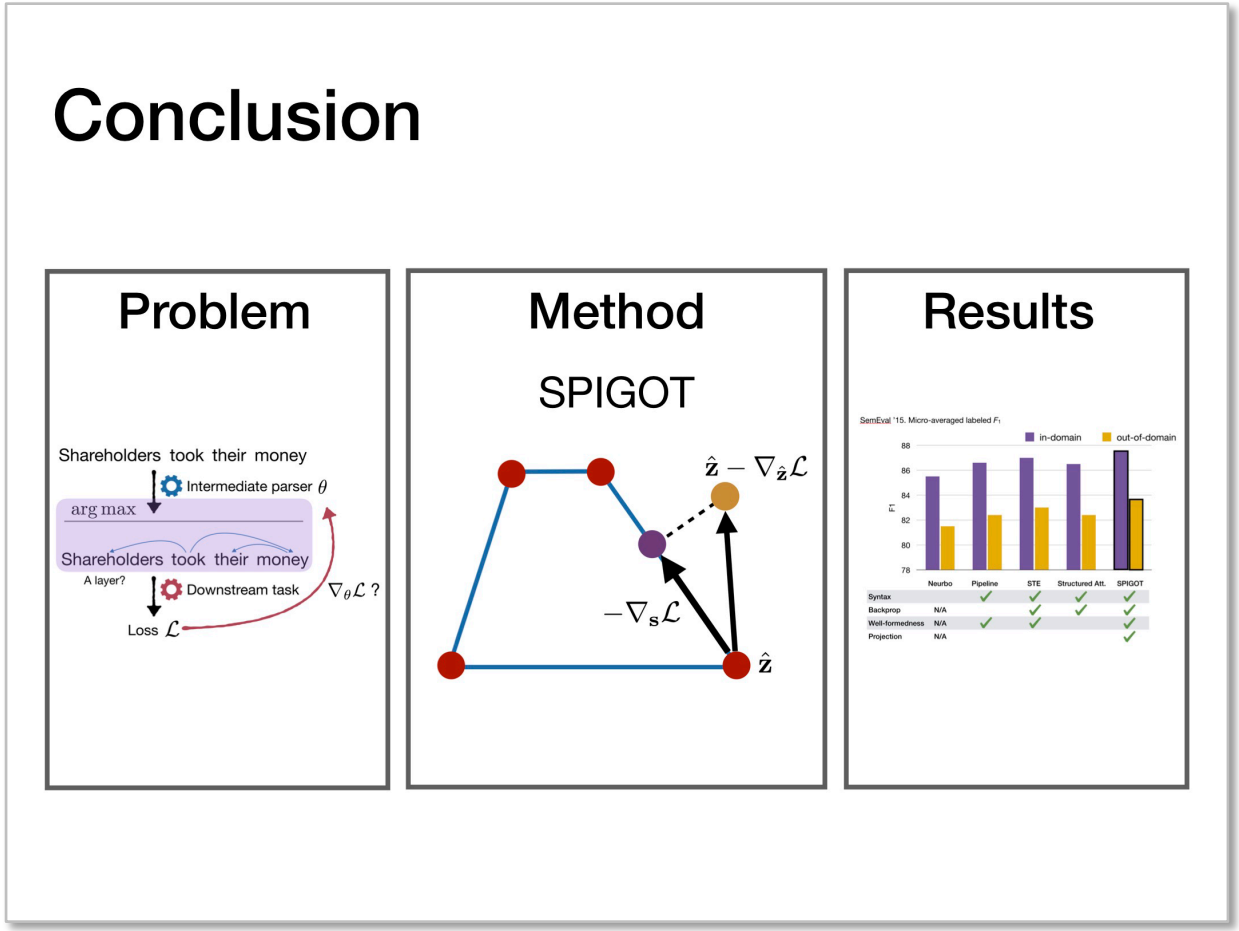
******

40

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

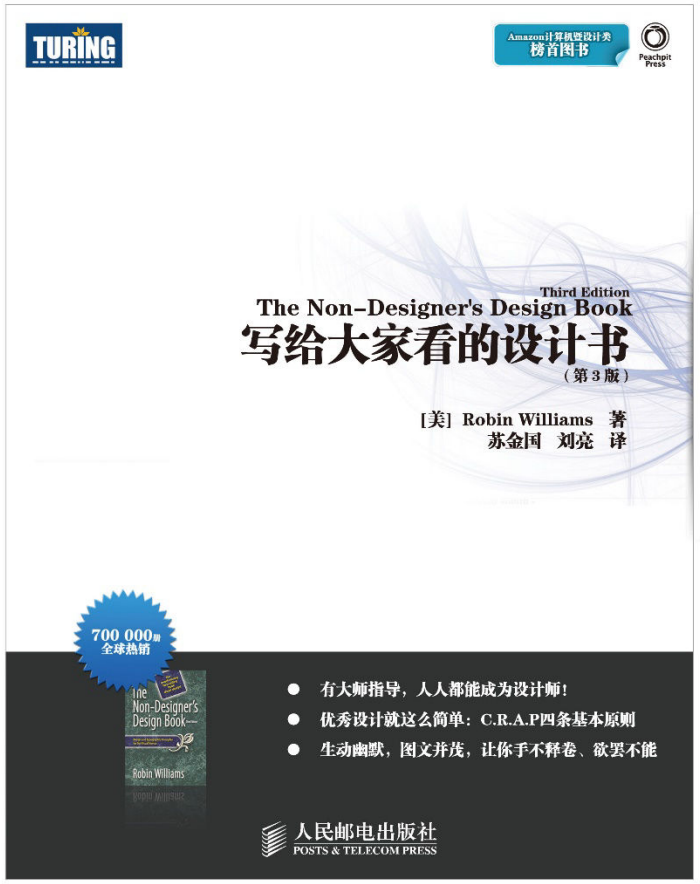
结论部分：新的展现形式



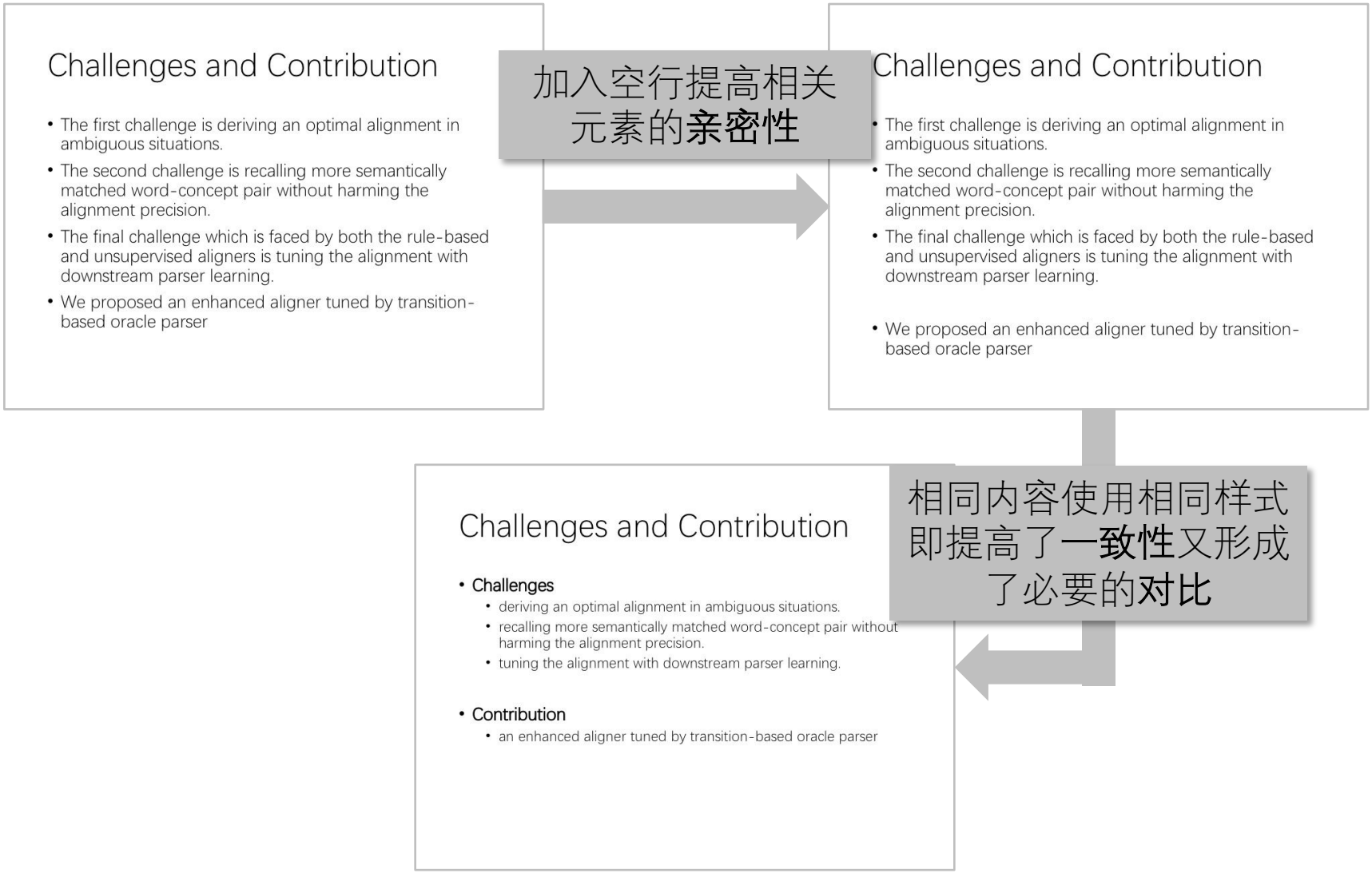
Hao Peng, Sam Thomson, and Noah A. Smith. 2018. Backpropagating through Structured Argmax using a SPIGOT，最后一页

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 $\mathcal{A}$.

for $c \in C$ **do**
  $A_c \leftarrow \emptyset$;
for $\rho_M \in \mathrm{P}_M$ **do**
  for $w_{s,e} \leftarrow$ *spans*(W) **do**
    for $c \in C$ **do**
      if $\rho_M(c, w_{s,e})$ **then**
        $A_c \leftarrow A_c \cup (s, e, \mathrm{nil})$;
updated $\leftarrow$ true ;
while *updated is true* **do**
  updated $\leftarrow$ false;
  for $\rho_U \in \mathrm{P}_U$ **do**
    for $c, c' \in C \times C$ **do**
      for $(s, e, d) \in A'_c$ **do**
        if $\rho_U(c, w_{s,e}) \wedge (s, e, c') \notin A_c$ **then**
          $A_c \leftarrow A_c \cup (s, e, c')$;
          updated $\leftarrow$ true;
$\mathcal{A} \leftarrow \emptyset$;
for $(a_1, ..., a_c) \in$ *CartesianProduct*$(A_1, ..., A_{|C|})$ **do**
  legal $\leftarrow$ true;
  for $a \in (a_1, ..., a_c)$ **do**
    $(s, e, c') \leftarrow a$;
    $(s', e', d) \leftarrow a_{c'}$;
    if $s \neq s' \wedge e \neq e'$ **then**
      legal $\leftarrow$ false ;
  if *legal* **then**
    $\mathcal{A} \leftarrow \mathcal{A} \cup (a_1, ..., a_c)$;

“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

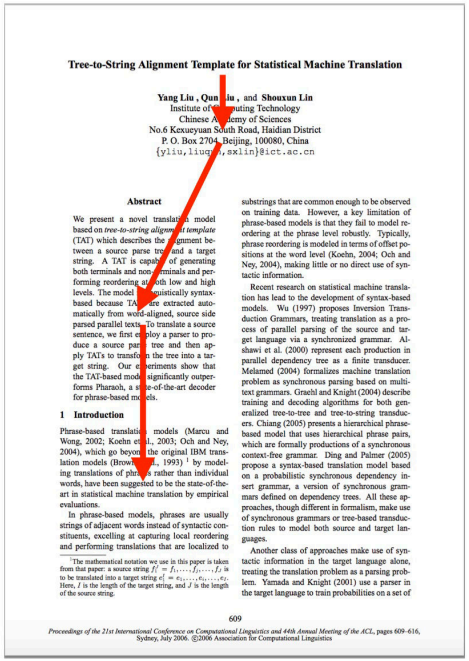
Aligner	Alignment F1 (on hand-align)	Oracle's Smatch (on dev. dataset)
JAMR	90.6	91.7
Our	95.2	94.7

- 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

信息流的变化



Tree-to-String Alignment Template for Statistical Machine Translation

Yang Liu , Qun Liu , and Shouxun Lin
Institute of Computing Technology
Chinese Academy of Sciences
No.6 Kexueyuan South Road, Haidian District
P. O. Box 2704, Beijing, 100080, China
{yliu, liuqun, sxlin}@ict.ac.cn

Abstract

We present a novel translation model based on *tree-to-string alignment template* (TAT) which describes the alignment between a source parse tree and a target string. A TAT is capable of generating both terminals and non-terminals and performing reordering at both low and high levels. The model is linguistically syntax-based because TATs are extracted automatically from word-aligned, source side parsed parallel texts. To translate a source sentence, we first employ a parser to produce a source parse tree and then apply TATs to transform the tree into a target string. Our experiments show that the TAT-based model significantly outperforms Pharaoh, a state-of-the-art decoder for phrase-based models.

1 Introduction

Phrase-based translation models (Marcu and Wong, 2002; Koehn et al., 2003; Och and Ney, 2004), which go beyond the original IBM translation models (Brown et al., 1993) [1] by modeling translations of phrases rather than individual words, have been suggested to be the state-of-the-art in statistical machine translation by empirical evaluations.

In phrase-based models, phrases are usually strings of adjacent words instead of syntactic constituents, excelling at capturing local reordering and performing translations that are localized to substrings that are common enough to be observed on training data. However, a key limitation of phrase-based models is that they fail to model reordering at the phrase level robustly. Typically, phrase reordering is modeled in terms of offset positions at the word level (Koehn, 2004; Och and Ney, 2004), making little or no direct use of syntactic information.

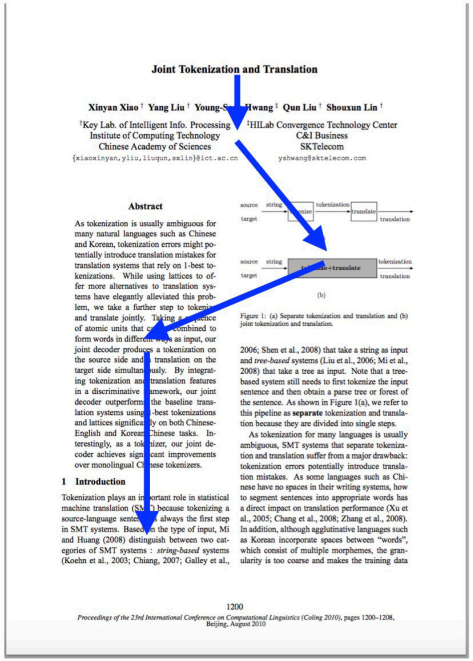
Recent research on statistical machine translation has lead to the development of syntax-based models. Wu (1997) proposes Inversion Transduction Grammars, treating translation as a process of parallel parsing of the source and target language via a synchronized grammar. Alshawi et al. (2000) represent each production in parallel dependency tree as a finite transducer. Melamed (2004) formalizes machine translation problem as synchronous parsing based on multi-text grammars. Graehl and Knight (2004) describe training and decoding algorithms for both generalized tree-to-tree and tree-to-string transducers. Chiang (2005) presents a hierarchical phrase-based model that uses hierarchical phrase pairs, which are formally productions of a synchronous context-free grammar. Ding and Palmer (2005) propose a syntax-based translation model based on a probabilistic synchronous dependency insert grammar, a version of synchronous grammars defined on dependency trees. All these approaches, though different in formalism, make use of synchronous grammars or tree-based transduction rules to model both source and target languages.

Another class of approaches make use of syntactic information in the target language alone, treating the translation problem as a parsing problem. Yamada and Knight (2001) use a parser in the target language to train probabilities on a set of

[1] The mathematical notation we use in this paper is taken from that paper: a source string $f_1^J = f_1, \ldots, f_j, \ldots, f_J$ is to be translated into a target string $e_1^I = e_1, \ldots, e_i, \ldots, e_I$. Here, I is the length of the target string, and J is the length of the source string.

609

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 609–616, Sydney, July 2006. ©2006 Association for Computational Linguistics



Joint Tokenization and Translation

Xinyan Xiao [†] Yang Liu [†] Young-Sook Hwang [‡] Qun Liu [†] Shouxun Lin [†]

[†]Key Lab. of Intelligent Info. Processing
Institute of Computing Technology
Chinese Academy of Sciences
{xiaoxinyan,yliu,liuqun,sxlin}@ict.ac.cn

[‡]HLLab Convergence Technology Center
C&I Business
SKTelecom
yshwang@sktelecom.com

Abstract

As tokenization is usually ambiguous for many natural languages such as Chinese and Korean, tokenization errors might potentially introduce translation mistakes for translation systems that rely on 1-best tokenizations. While using lattices to offer more alternatives to translation systems have elegantly alleviated this problem, we take a further step to tokenize and translate jointly. Taking a sequence of atomic units that can be combined to form words in different ways as input, our joint decoder produces a tokenization on the source side and a translation on the target side simultaneously. By integrating tokenization and translation features in a discriminative framework, our joint decoder outperforms the baseline translation systems using 1-best tokenizations and lattices significantly on both Chinese-English and Korean-Chinese tasks. Interestingly, as a tokenizer, our joint decoder achieves significant improvements over monolingual Chinese tokenizers.

1 Introduction

Tokenization plays an important role in statistical machine translation (SMT) because tokenizing a source-language sentence is always the first step in SMT systems. Based on the type of input, Mi and Huang (2008) distinguish between two categories of SMT systems : *string-based* systems (Koehn et al., 2003; Chiang, 2007; Galley et al., 2006; Shen et al., 2008) that take a string as input and *tree-based* systems (Liu et al., 2006; Mi et al., 2008) that take a tree as input. Note that a tree-based system still needs to first tokenize the input sentence and then obtain a parse tree or forest of the sentence. As shown in Figure 1(a), we refer to this pipeline as **separate** tokenization and translation because they are divided into single steps.

As tokenization for many languages is usually ambiguous, SMT systems that separate tokenization and translation suffer from a major drawback: tokenization errors potentially introduce translation mistakes. As some languages such as Chinese have no spaces in their writing systems, how to segment sentences into appropriate words has a direct impact on translation performance (Xu et al., 2005; Chang et al., 2008; Zhang et al., 2008). In addition, although agglutinative languages such as Korean incorporate spaces between "words", which consist of multiple morphemes, the granularity is too coarse and makes the training data

Figure 1: (a) Separate tokenization and translation and (b) joint tokenization and translation.

1200

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1200–1208, Beijing, August 2010

43

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

(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	560
	Our Parser (greedy)	92.0 (+2.1)	1013 (×1.8)
	Zpar: beam = 64	92.9*	29*
Graph -based	MSTParser	92.0	12
	TurboParser	93.1*	31*

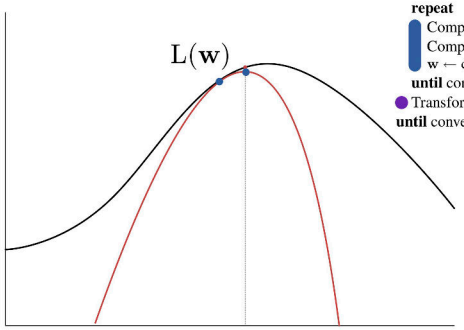
A Fast and Accurate Dependency Parser using Neural Networks 3

为了抓住听众，把最好的部分前置

Berkeley NLP

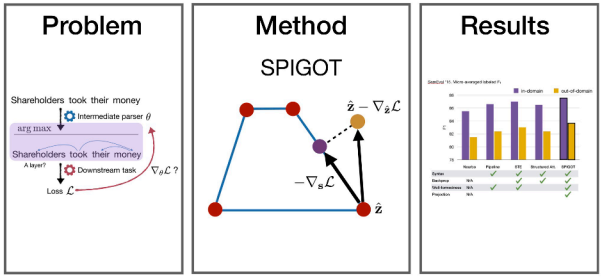
EM with Features

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



模型部分有取舍，用好图和例子

Conclusion



“结论” 也有新思路

Challenges and Contrib

- Challenges
  - deriving an optimal alignment in ambiguous s
  - 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-base

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