Automatically Acquired Lexical Knowledge Improves Japanese Joint Morphological and Dependency Analysis

Daisuke Kawahara  Yuta Hayashibe*1  Hajime Morita*2  Sadao Kurohashi*

Kyoto University  *JST CREST
1 Current affiliation is Fairy Devices Inc.  2 Current affiliation is Fujitsu Laboratories Ltd.

IWPT 2017 (September 21, 2017)
Knowledge Acquisition and Knowledge-based NLP

Mary ate the salad.

Mary ate the salad.

**Case frames**

泳ぐ swim

{ 人 person, 子 child,...} が
{クロール crawl, 平泳ぎ,...} で
{海 sea, 大海,...} を

見る see

{ 人 person, 者,...} が
{望遠鏡 telescope, 双眼鏡,...} で
{姿 figure, 人 person,...} を

クロールで 水中で泳いでいる 女の子を 見た

crawl  swim  girl  saw

望遠鏡で 水中で泳いでいる 女の子を 見た

telecope  swim  girl  saw
We Need to Segment a Sentence!

クロールで泳いでいる女の子を見た
crawl swim girl saw

クロールで泳いでいる女の子を見た
crawl swim girl saw
We Need to Segment a Sentence!

• Word segmentation is necessary before applying dependency parsing for unsegmented languages, such as Chinese and Japanese
• Such pipeline framework causes the problem of error propagation
• Several supervised joint models have achieved some success for Chinese but not for Japanese

Question: Can lexical knowledge improve Japanese joint morphological and dependency analysis?
可能性があるかないか分からないか分からないか

I don’t know whether there is a possibility or
I don’t know that a possibility doesn’t walk
Related work (1/2)

• Joint transition-based parsing
  – POS tagging and parsing [Bohnet+, 2013] [Wang+, 2014]
  – Chinese word segmentation, POS tagging and dependency parsing [Hatori+, 2012] [Zhang+, 2014] [Kurita+, 2017]

• Lattice parsing [Goldberg+, 2009] [Green+, 2010] [Goldberg+, 2011]
Related work (2/2)

• Dependency parsing models using lexical knowledge [van Noord, 2007] [Koo+, 2008] [Chen+, 2009] [Bansal+, 2011]

• Japanese dependency parsing models
  – Transition-based (supervised) models [Kudo+, 2002] [Sassano, 2004] [Yoshinaga+., 2014]
  – Probabilistic model based on case frames [Kawahara+., 2006]
Lexical Knowledge

- Case frames
- Cooccurrence probabilities of noun-noun / predicate-predicate dependencies
- Word embeddings

<table>
<thead>
<tr>
<th>Case Frame</th>
<th>ある (exist):3</th>
</tr>
</thead>
<tbody>
<tr>
<td>が (NOM)</td>
<td>possibility: 121867</td>
</tr>
<tr>
<td>に (DAT)</td>
<td>price: 23, myself: 20, you: 18, ...</td>
</tr>
<tr>
<td>で (LOC)</td>
<td>step: 4, influence: 4, ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case Frame</th>
<th>あるく (walk):1</th>
</tr>
</thead>
<tbody>
<tr>
<td>が (NOM)</td>
<td>person: 57, l: 13, ...</td>
</tr>
<tr>
<td>を (ACC)</td>
<td>road: 24236, trail: 4066, ...</td>
</tr>
<tr>
<td>から (ABL)</td>
<td>parking: 175, station: 88, ...</td>
</tr>
</tbody>
</table>
Case Frame Compilation

- Web
  - 10G sentences (3G pages)
  - Parsing and Filtering
    - 89.0% for all
    - 98.3% for 20.7% P-As
  - Predicate-Argument structures (PAS)
    - Clustering
  - Case frames for 120K predicates

[Kawahara+, 2006] [Kawahara+, 2014]
## Case frame examples for *tsumu* (積む)

<table>
<thead>
<tr>
<th></th>
<th>CS</th>
<th>instances (translated into English)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tsumu (1)</strong> (accumulate experience)</td>
<td><em>ga</em></td>
<td>player:21, all:20, person:142, …</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>wo</em> experience:100127, achievement:10350, …</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>de</em> site:240, area:209, …</td>
</tr>
<tr>
<td><strong>tsumu (2)</strong> (pursue/devote)</td>
<td><em>ga</em></td>
<td>person:27, player:13, all:12, …</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>wo</em> exercise:15579, study:13222, …</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>de</em> basis:694, under:384, university:99, …</td>
</tr>
<tr>
<td><strong>tsumu (3)</strong> (load)</td>
<td><em>ga</em></td>
<td>man:33, person:20, child:11, …</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>wo</em> baggage:11294, luggage:2989, …</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>ni</em> car:920, truck:160, bike:114, …</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Robust Case Frame Compilation

[Hayashibe+, 2015]

10G Japanese web sentences

Company NOM Scale ACC Enlarge

Scale ACC Company NOM Enlarge

Company’s Scale NOM Enlarge

Conventional Case Frames

Clustered

Transitive usages

Intransitive usages w/ inanimate nominatives

Intransitive usages w/o inanimate nominatives

New Case Frames

Enlarge

Case Arguments

NOM company, scale, …

ACC scale, area, …

DAT Japan, …

Case Arguments

NOM Scale, …

DAT Japan, …

Case Arguments

NOM member, …

DAT Double, …
Lexical Knowledge

• Case frames

• Cooccurrence probabilities of noun-noun / predicate-predicate dependencies
  – Calculate $P(predicate_1|predicate_2)$ and $P(noun_1|noun_2)$ from automatic parses

• Word embeddings [Mikolov+, 2013]
  – Clues for coordinate structures
Parsing Model

• Using the well-known CKY algorithm
• Procedure
  1. Project candidate words onto the CKY table
  2. Generate base phrases
     • a base phrase = a content word + 0 or more function words
  3. Generate dependencies and calculate their scores based on lexical knowledge
可能性があるかいないか（whether a possibility exists）
or
(a possibility doesn’t walk)
1. Project candidate words onto the CKY table

Input: 可能性があるかないか

<table>
<thead>
<tr>
<th>可</th>
<th>可能</th>
</tr>
</thead>
<tbody>
<tr>
<td>能</td>
<td></td>
</tr>
<tr>
<td>性</td>
<td></td>
</tr>
<tr>
<td>が</td>
<td>exist</td>
</tr>
<tr>
<td>有る</td>
<td>walk</td>
</tr>
<tr>
<td>歩か</td>
<td></td>
</tr>
<tr>
<td>る</td>
<td></td>
</tr>
<tr>
<td>か</td>
<td></td>
</tr>
<tr>
<td>ない</td>
<td></td>
</tr>
<tr>
<td>い</td>
<td></td>
</tr>
<tr>
<td>か</td>
<td></td>
</tr>
</tbody>
</table>
Input: 可能性があるかないか

1. Project candidate words onto the CKY table
2. Generate base phrases
   - a base phrase = a content word + 0 or more function words

possibility NOM
doesn’t walk

1. Project candidate words onto the CKY table
2. Generate base phrases
   - a base phrase = a content word + 0 or more function words
1. Project candidate words onto the CKY table
2. Generate base phrases
   • a base phrase = a content word + 0 or more function words
3. Generate dependencies and calculate their scores based on lexical knowledge
Features (1/2)

• Word feature
  – Marginal score of morphological analysis
• Base phrase features
  – Word 2,3-grams in a base phrase
  – # of base phrases in a sentence
  – Words at a base phrase boundary
  – # of predicates in a sentence
  – A predicate representation
• Dependency features
  – A dependency label
  – Content/function words and punctuations of a modifier
  – Content/function words and punctuations of a head
  – Distance between a modifier and its heads
Features (2/2)

• Features derived from lexical knowledge
  – # of predicates that do not have case frames
  – Probabilities calculated based on case frames
    • e.g., case frame/slot generating probability
  – A cooccurrence probability between nouns
  – A cooccurrence probability between predicates
  – Content word similarity between a modifier and its head
  – Similarity of word sequences for coordination
Experimental Settings (1/2)

• Dependency treebank
  – Kyoto Univ. Text Corpus (NEWS)
  – Kyoto Univ. Web Document Leads Corpus (WEB)

• Dependency unit
  – Base phrase dependencies

• Input of the parser
  – N-best output of the Japanese morphological analyzer JUMAN++ [Morita+, 2015]
JUMAN++: RNN-based Japanese Morphological Analyzer

• Recurrent Neural Network Language Model [Mikolov+, 2010]
  – A neural network based language model, with a hidden context layer
  – The model can calculate $p(w|\text{context})$ based on semantically generalized vector representation
Experimental Settings (1/2)

• Dependency treebank
  – Kyoto Univ. Text Corpus (NEWS)
  – Kyoto Univ. Web Document Leads Corpus (WEB)

• Dependency unit
  – Base phrase dependencies

• Input of the parser
  – N-best output of the Japanese morphological analyzer JUMAN++ [Morita+, 2015]
    – Apply 10-way jackknififfing to the training set

• Training of the parser
  – L-BFGS with L1 regularization

• Using beam search
  – Beam width = 10
Experimental Settings (2/2)

• Baseline for word segmentation and POS tagging
  – JUMAN++ [Morita+, 2015] (1-best)

• Baselines for dependency parsing
  – KNP [Kawahara+, 2006]
  – CaboCha (using the transition-based algorithm of [Sassano, 2004])
  – KNP+CaboCha
    • Base phrase chunking by KNP and dependency parsing by CaboCha
  – Our model without lexical knowledge (LK)
Results

**NEWS**

- Seg: 99.6, 99.4, 99.2, 99, 98.8, 98.6, 98.4, 98.2, 98
- Seg+POS: 98.2, 98.4, 98.6, 98.8, 99, 99.2, 99.4, 99.6

**WEB**

- Seg: 96.6, 98.4, 98.2, 98, 97.8, 97.6
- Seg+POS: 98.2, 98.4, 98.6, 98.8, 99, 99.2, 99.4, 99.6

---

- JUMAN++ (1-best)
- KNP++ (N-best)

---

- Dep UAS
  - KNP+CaboCha
  - KNP++ (1-best)
  - KNP++ (N-best) wo/LK
  - KNP++ (N-best)

---

(F1)
Improved Examples

- 雑音対策 (noise control)
- 課題 (task)
- COPULA
- てもの (bargain)
- もの (thing)
- トップ (TOP)
- 店 (shop)
- は (that)
- あの (nephew)
- お (nasty)
- や (and)
- おい (prefix)
- 見つかる (found)
- よく (often)
- が (that)
- が (NOM)
- 研究 (research)
- が (NOM)
- 必要 (necessary)
- で (of)
- COPULA
- で (in)
- が (which)
- て (in)
- で (in)
- で (in)
- お (nephew)
- お (nasty)
- と (and)
- い (nasty)
- CMI (CMI)
- 別れた (parted)

✓ 表示が正しい
✗ 表示が間違っている
Discussion

• The 1-best accuracy of segmentation and POS tagging is already very high, especially for NEWS
  – However, we can improve it by reranking N-best outputs based on lexical knowledge, especially for WEB

• The gold does not distinguish some ambiguous cases
Summary

• Automatically acquired lexical knowledge actually improved Japanese joint morphological and dependency analysis!
• We will release lexical resources and analyzers
  – RNN-based Japanese morphological analyzer (JUMAN++)
  – Case frames compiled from 10G Japanese sentences
  – Joint Japanese morphological and syntactic analyzer based on lexical knowledge (KNP++)
Future Work

• Neuralize it!
• Integrate PAS analysis (including zero anaphora resolution) into our joint morphological and syntactic analysis