Reducing Approximation and Estimation Errors for Chinese Lexical Processing with Heterogeneous Annotations

Weiwei Sun and Xiaojun Wan

Saarland University
LT-lab, DFKI
Peking University

July 9, 2012
1-minute talk

- Many NLP systems rely on large-scale, manually annotated corpora.
- Linguistic annotations are
  - important to train statistical models
  - very expensive to build
- Multiple heterogeneous annotations EXIST!
  - Parsing: Penn Treebank vs. Redwoods Treebank
  - Semantic role labeling: Propbank vs. FrameNet
- Different projects $\rightarrow$ different linguistic theories $\rightarrow$ different annotation schemes
- How to consume heterogeneous annotations?
- Annotation ensemble for Chinese lexical processing.
Outline

Annotation ensemble

Joint word segmentation and POS tagging

A sub-word tagging model

Experiments
Outline

Annotation ensemble

Joint word segmentation and POS tagging

A sub-word tagging model

Experiments
Annotation ensemble

How to consume heterogeneous annotations?
Annotation ensemble

How to consume heterogeneous annotations?

Two essential characteristics

1. Heterogeneous annotations are (similar but) different.
   - Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.

2. Heterogeneous annotations are (different but) similar.
   - Same high-level linguistic principles.
Annotation ensemble

How to consume heterogeneous annotations?

Two essential characteristics

1. Heterogeneous annotations are different.
   • Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.
   • Reducing approximation errors

2. Heterogeneous annotations are similar.
   • Same high-level linguistic principles.
   • Reducing estimation errors

• The approximation error: intrinsic suboptimality of the model
• The estimation error: having only finite training data
Annotation ensemble

How to consume heterogeneous annotations?

Two essential characteristics

1. Heterogeneous annotations are *different*.
   - Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.

2. Heterogeneous annotations are *similar*.
   - Same high-level linguistic principles.

• The approximation error: intrinsic suboptimality of the model
• The estimation error: having only finite training data
## Annotation ensemble

### How to consume heterogeneous annotations?

**Two essential characteristics**

1. Heterogeneous annotations are *different*.
   - Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.

2. Heterogeneous annotations are *similar*.
Annotation ensemble

How to consume heterogeneous annotations?

Two essential characteristics

1. Heterogeneous annotations are different.
   ○ Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.

2. Heterogeneous annotations are similar.
   ○ Same high-level linguistic principles.
Annotation ensemble

How to consume heterogeneous annotations?

Two essential characteristics

1. Heterogeneous annotations are (similar but) different.
   - Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.

2. Heterogeneous annotations are (different but) similar.
   - Same high-level linguistic principles.
Annotation ensemble

How to consume heterogeneous annotations?

Two essential characteristics

1. Heterogeneous annotations are (similar but) different.
   - Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.

2. Heterogeneous annotations are (different but) similar.
   - Same high-level linguistic principles.

- The approximation error: intrinsic suboptimality of the model
- The estimation error: having only finite training data
Annotation ensemble

How to consume heterogeneous annotations?

Two essential characteristics

1. Heterogeneous annotations are (similar but) different.
   ◦ Different projects, different linguistic theories, different representation formalisms, different annotation schemes, etc.
   ☺ Reducing approximation errors

2. Heterogeneous annotations are (different but) similar.
   ◦ Same high-level linguistic principles.
   ☺ Reducing estimation errors

- The approximation error: intrinsic suboptimality of the model
- The estimation error: having only finite training data
Previous work on Chinese syntactic parsing

Reducing approximation error: Stacking (Parse reranking)

- A parser trained on Treebank $\mathcal{D}_A$ produce $A$-style $n$-best parses $\mathcal{L}_n^A$.
- A parser trained on Treebank $\mathcal{D}_B$ produce an auxiliary $B$-style parse $\mathcal{L}_1^B$.
- Reranking $\mathcal{L}_n^A$ with complementary features extracted from $\mathcal{L}_1^B$.

Reducing estimation error: Treebank converting

- Convert Treebank $\mathcal{D}_B$ to its correlated $A$-style parses $\mathcal{D}_{B \rightarrow A}$.
  - For treebank conversion, linguistic rules are popular.
- Train a new $A$-style parser with $\mathcal{D}_A \cup \mathcal{D}_{B \rightarrow A}$. 
Previous work on Chinese word segmentation and POS tagging

Reducing approximation error: Stacking

Training:
- By using the corpus $D_B$, train an auxiliary $B$-style tagger $T_B$.
- By applying $T_B$, label the corpus $D_A$ and get a new $\hat{B}$-augmented corpus $D_A^{\hat{B}} = \{\langle s^{(1)}, \hat{b}^{(1)}, a^{(1)} \rangle \ldots \}$.
- By using $D_A^{\hat{B}}$, train a $\hat{B}$-augmented tagger $T_A^{\hat{B}}$.

Prediction:
- $T_B$ is used to produce an auxiliary $B$-style analysis $L_1^B$.
- Tagging by using $T_A^{\hat{B}}$ with features extracted from $L_1^B$. 
A general framework for annotation ensemble

- Inference: new features, new structures ⇒ reducing the approximation error.
- Inference: as a corpus conversion procedure ⇒ increasing reliable training data ⇒ reducing the estimation error.
A general framework for annotation ensemble

- Inference: new features, new structures ⇒ reducing the approximation error.
- Inference: as a corpus conversion procedure ⇒ increasing reliable training data ⇒ reducing the estimation error.
A general framework for annotation ensemble

- Inference: new features, new structures $\Rightarrow$ reducing the approximation error.
- Inference: as a corpus conversion procedure $\Rightarrow$ increasing reliable training data $\Rightarrow$ reducing the estimation error.
Outline

Annotation ensemble

Joint word segmentation and POS tagging

A sub-word tagging model

Experiments
Joint word segmentation and POS tagging

Task:

- Input: a sequence of characters
- Output: a sequence of \( \langle \text{start, end, tag} \rangle \) tuples
- Labeled segmentation
- Like text chunking
Joint word segmentation and POS tagging

Example

Input: a sequence of characters
Joint word segmentation and POS tagging

Example

Segmentation and tagging according to colors
Joint word segmentation and POS tagging

Example

Segmentation and tagging according to CONTENT
Joint word segmentation and POS tagging

Example

Segmentation and tagging according to colors
Two representative corpora

Two corpora for Chinese lexical processing:
- Chinese Treebank (CTB)
  - Treebank-driven
    - Word: terminals of a constituent tree
    - POS: pre-terminals of a constituent tree
- PKU’s People’s Daily data (PPD)
  - Lexicon-driven
    - Instances of lexicon entries

Manually label 200 CTB sentences according to the PPD standard.
- aclweb.org/supplementals/P/P12/P12-1025.Datasets.zip

Two corpora are systematically different but highly compatible.
Key properties

Word segmentation:
- 90%+ words have same boundaries.
- Among 38K+ words, only one cross-bracketing occurs.
  - If $ABC$ is segmented as $[AB]C$ in one corpus,
  - it is highly unlikely that $ABC$ is segmented as $A[BC]$ in the other.

POS tagging:
- CTB: treebank-driven
  - syntactic/dynamic properties.
- PPD: lexicon-driven
  - lexical/static properties.
Examples for POS annotation

CTB’s common verbs (VV) are

• mainly labeled as verbs (v) in PPD,
• sometimes labeled as nominal categories (a, vn, n) in PPD.
Examples for POS annotation

CTB’s **common verbs** (VV) are
- mainly labeled as **verbs** (v) in PPD,
- sometimes labeled as **nominal categories** (a, vn, n) in PPD.

**Why?**
Examples for POS annotation

CTB’s common verbs (VV) are
• mainly labeled as verbs (v) in PPD,
• sometimes labeled as nominal categories (a, vn, n) in PPD.

Why?
There are a large number of Chinese adjectives and nouns that can be realized as predicates without linking verbs.
Examples for POS annotation

CTB’s common nouns (NN) are

- mainly labeled as nouns (n) in PPD,
- sometimes labeled as verbal categories (vn, v) in PPD.
Examples for POS annotation

CTB’s **common nouns** (NN) are
- mainly labeled as **nouns (n)** in PPD,
- sometimes labeled as **verbal categories (vn, v)** in PPD.

**Why?**
Examples for POS annotation

CTB’s common nouns (NN) are

- mainly labeled as nouns (n) in PPD,
- sometimes labeled as verbal categories (vn, v) in PPD.

Why?

A majority of Chinese verbs could be realized as subjects and objects without form changes.
Outline

Annotation ensemble

Joint word segmentation and POS tagging

A sub-word tagging model

Experiments
State-of-the-art

Single view solution:
- Character-based vs. word-based
- Markov tagging vs. semi-Markov tagging
- Not bound to specific annotations

Previous work on annotation ensemble:
- **Feature-based** stacking
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complimentary features.
- The approximation error is reduced

Example
State-of-the-art

Single view solution:

- **Character-based** vs. **word-based**
- **Markov tagging** vs. **semi-Markov tagging**
- Not bound to specific annotations

Previous work on annotation ensemble:

- **Feature-based stacking**
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complimentary features.

- The approximation error is reduced

Example
State-of-the-art

Single view solution:

- **Character-based** vs. word-based
- **Markov tagging** vs. semi-Markov tagging
- Not bound to specific annotations

Previous work on annotation ensemble:

- **Feature-based stacking**
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complimentary features.

- The approximation error is reduced

Example

...
State-of-the-art

Single view solution:

- **Character-based** vs. **word-based**
- **Markov tagging** vs. **semi-Markov tagging**
- Not bound to specific annotations

Previous work on annotation ensemble:

- **Feature-based stacking**
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complimentary features.
- The approximation error is reduced

Example

刘 刘 刘
B-nr B-NR B-NR
华 华 华
B-nr I-NR I-NR
清 清 清
I-nr I-NR I-NR
副 副 副
B-b B-NN B-NN
总 总 总
B-n I-NN I-NN
理 理 理
I-n I-NN I-NN
的 的 的
B-u B-DEG B-DEG
这 这 这
B-r B-DT B-DT
次 次 次
I-r B-M B-M
访 访 访
I-v I-NN I-NN
...
State-of-the-art

Single view solution:

- **Character-based** vs. **word-based**
- **Markov tagging** vs. semi-Markov tagging
- Not bound to specific annotations

Previous work on annotation ensemble:

- **Feature-based stacking**
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complementary features.
- The approximation error is reduced

Example

刘 刘 刘
  B-nr B-NR B-nr

华 华 华
  B-nr I-NR I-nr

清 清 清
  I-nr I-NR I-nr

副 副 副
  B-b B-NN B-b

总 总 总
  B-n I-NN I-n

理 理 理
  I-n I-NN I-n

的 的 的
  B-u B-DEG B-u

这 这 这
  B-r B-DT B-r

次 次 次
  I-r B-M I-r

来 来 来
  B-v B-NN B-v

访 访 访
  I-v I-NN I-v

...
State-of-the-art

Single view solution:

- **Character-based** vs. **word-based**
- **Markov tagging** vs. **semi-Markov tagging**
- Not bound to specific annotations

Previous work on annotation ensemble:

- **Feature-based stacking**
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complimentary features.
- The approximation error is reduced

Example
State-of-the-art

Single view solution:

- **Character-based** vs. **word-based**
- **Markov tagging** vs. **semi-Markov tagging**
- **Not bound to specific annotations**

Previous work on annotation ensemble:

- **Feature-based stacking**
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complimentary features.
- The approximation error is reduced

Example

```
刘 刘 刘 B-NR
华 华 华 I-NR
清 清 清 I-NR
副 副 副 B-NN
总 总 总 I-NN
理 理 理 I-NN
的 的 的 B-u
这 这 这 B-r
次 次 次 B-M
来 来 来 B-NN
访 访 访 I-NN
... ...
```
State-of-the-art

Single view solution:

• **Character-based** vs. **word-based**

• **Markov tagging** vs. **semi-Markov tagging**

• Not bound to specific annotations

Previous work on annotation ensemble:

• **Feature-based** stacking
  
  ○ Auxiliary solvers are trained on heterogeneous annotations.
  
  ○ Auxiliary solvers are employed to produce **complementary features**.
  
  ○ Target solver utilizes these **complimentary features**.

• The approximation error is reduced

**Example**

刘 华 清 副 总 理 的 这 次 来 访

...
State-of-the-art

Single view solution:

- **Character-based** vs. **word-based**
- **Markov tagging** vs. **semi-Markov tagging**
- Not bound to specific annotations

Previous work on annotation ensemble:

- **Feature-based** stacking
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complementary features.
- The approximation error is reduced
State-of-the-art

Single view solution:

- **Character-based** vs. **word-based**
- **Markov tagging** vs. semi-Markov tagging
- Not bound to specific annotations

Previous work on annotation ensemble:

- **Feature-based** stacking
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce **complementary features**.
  - Target solver utilizes these complementary features.
- The approximation error is reduced

Example

| 刘 | B-nr |
| 华 | B-nr |
| 清 | l-nr |
| 副 | B-b  |
| 总 | B-n  |
| 理 | l-n  |
| 的 | B-u  |
| 这 | B-r  |
| 次 | l-r  |
| 来 | B-v  |
| 访 | l-v  |
| ... | ... |
State-of-the-art

Single view solution:
- **Character-based** vs. word-based
- **Markov tagging** vs. semi-Markov tagging
- Not bound to specific annotations

Previous work on annotation ensemble:
- **Feature-based stacking**
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complementary features.
- The approximation error is reduced

Example

<table>
<thead>
<tr>
<th>刘</th>
<th>B-nr</th>
</tr>
</thead>
<tbody>
<tr>
<td>华</td>
<td>B-nr</td>
</tr>
<tr>
<td>清</td>
<td>I-nr</td>
</tr>
<tr>
<td>副</td>
<td>B-b</td>
</tr>
<tr>
<td>总</td>
<td>B-n</td>
</tr>
<tr>
<td>理</td>
<td>I-n</td>
</tr>
<tr>
<td>的</td>
<td>B-u</td>
</tr>
<tr>
<td>这</td>
<td>B-r</td>
</tr>
<tr>
<td>次</td>
<td>I-r</td>
</tr>
<tr>
<td>来</td>
<td>B-v</td>
</tr>
<tr>
<td>访</td>
<td>I-v</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
State-of-the-art

Single view solution:

- **Character-based** vs. word-based
- **Markov tagging** vs. semi-Markov tagging
- Not bound to specific annotations

Previous work on annotation ensemble:

- **Feature-based** stacking
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce *complementary features*.
  - Target solver utilizes these *complimentary features*.

- The approximation error is reduced

Example

| 刘 | B-nr |
| 华 | B-nr |
| 清 | I-nr |
| 副 | B-b   B-NN |
| 总 | B-n   |
| 理 | I-n   |
| 的 | B-u   |
| 这 | B-r   |
| 次 | I-r   |
| 来 | B-v   |
| 访 | I-v   |
| ... | ... |
State-of-the-art

Single view solution:

- **Character-based** vs. **word-based**
- **Markov tagging** vs. **semi-Markov tagging**
- Not bound to specific annotations

Previous work on annotation ensemble:

- **Feature-based** stacking
  - Auxiliary solvers are trained on heterogeneous annotations.
  - Auxiliary solvers are employed to produce complementary features.
  - Target solver utilizes these complementary features.
- The approximation error is reduced

Example

```
刘  B-nr  B-NR
华  B-nr  I-NR
清  I-nr  I-NR
副  B-b   B-NN
总  B-n   I-NN
理  I-n   I-NN
的  B-u   B-DEG
这  B-r   B-DT
次  I-r   B-M
来  B-v   B-NN
访  I-v   I-NN
...  ...  ...
```
A structure-based stacking model

Penn Chinese Treebank

PKU People's Daily

Character-based approach

Model

Inference via Sub-word tagging

Output

Model

Input
A structure-based stacking model

- Reducing approximation errors
  - Stacking [ Feature / structure ]

- Reducing estimation errors
  - Corpus conversion [ Stacking model is a statistical converter ]
  - Model retraining
Workflow
Workflow

System architecture

Raw sentences
Workflow

Heterogeneous solves

• trained on heterogeneously labeled data.
• Single view
Workflow

Structured sentences
- segmented
- tagged

System architecture

Raw sentences
- Heterogeneous solver A
  - Structured sentences
- Heterogeneous solver B
  - Structured sentences
- Heterogeneous solver C
  - Structured sentences
Merging:

- Maximizing agreements of non-word-breaks
- If two continuous characters are separated by any solver, it is taken as a sub-word break.
Sub-words are

- as large as possible
- compatible with all segmentation
Workflow

Good sub-word tagging
- good segmentation
- good POS tagging

System architecture

Raw sentences
- Heterogeneous solver A
  - Structured sentences
  - Merging
    - Sub-word sequences
    - Sub-word tagger SubTag
- Heterogeneous solver B
  - Structured sentences
- Heterogeneous solver C
  - Structured sentences
Workflow

Also a statistical corpus converter.
An example

Example

刘 华 清 副 总 理 的 这 次 来 访
An example

Example

刘 B-nr
华 B-nr
清 I-nr
副 B-b
总 B-n
理 I-n
的 B-u
这 B-r
次 I-r
来 B-v
访 I-v
An example

Example

刘 B-nr B-NR
华 B-nr I-NR
清 I-nr I-NR
副 B-b B-NN
总 B-n I-NN
理 I-n I-NN
的 B-u B-DEC
这 B-r B-DT
次 I-r B-M
来 B-v B-NN
访 I-v I-NN
An example

Example

刘 B-nr B-NR 刘
华 B-nr I-NR 华清
清 I-nr I-NR
副 B-b B-NN 副
总 B-n I-NN 总理
理 I-n I-NN
的 B-u B-DEC 的
这 B-r B-DT 这
次 I-r B-M 次
来 B-v B-NN 来访
访 I-v I-NN
An example

Example

刘  B-nr  B-NR  刘  B-nr  B-NR
华  B-nr  I-NR  华 清  B-nr  I-NR
清  I-nr  I-NR
副  B-b  B-NN  副  B-b  B-NN
总  B-n  I-NN  总 理  B-n  I-NN
理  I-n  I-NN
的  B-u  B-DEC  的  B-u  B-DEC
这  B-r  B-DT  这  B-r  B-DT
次  I-r  B-M  次  I-r  B-M
来  B-v  B-NN  来 访  B-v  B-NN
访  I-v  I-NN
An example

Example

刘 B-nr B-NR 刘 B-nr B-NR
华 B-nr I-NR 华清 B-nr I-NR
清 I-nr I-NR
副 B-b B-NN 副 B-b B-NN
总 B-n I-NN 总理 B-n I-NN
理 I-n I-NN
的 B-u B-DEC 的 B-u B-DEC
这 B-r B-DT 这 B-r B-DT
次 I-r B-M 次 I-r B-M
来 B-v B-NN 来访 B-v B-NN
访 I-v I-NN
An example

Example

<table>
<thead>
<tr>
<th>刘</th>
<th>B-nr</th>
<th>B-NR</th>
<th>刘</th>
<th>B-nr</th>
<th>B-NR</th>
</tr>
</thead>
<tbody>
<tr>
<td>华</td>
<td>B-nr</td>
<td>I-NR</td>
<td>华清</td>
<td>B-nr</td>
<td>I-NR</td>
</tr>
<tr>
<td>清</td>
<td>I-nr</td>
<td>I-NR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>副</td>
<td>B-b</td>
<td>B-NN</td>
<td>副</td>
<td>B-b</td>
<td>B-NN</td>
</tr>
<tr>
<td>总</td>
<td>B-n</td>
<td>I-NN</td>
<td>总理</td>
<td>B-n</td>
<td>I-NN</td>
</tr>
<tr>
<td>理</td>
<td>I-n</td>
<td>I-NN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>的</td>
<td>B-u</td>
<td>B-DEC</td>
<td>的</td>
<td>B-u</td>
<td>B-DEC</td>
</tr>
<tr>
<td>这</td>
<td>B-r</td>
<td>B-DT</td>
<td>这</td>
<td>B-r</td>
<td>B-DT</td>
</tr>
<tr>
<td>次</td>
<td>I-r</td>
<td>B-M</td>
<td>次</td>
<td>I-r</td>
<td>B-M</td>
</tr>
<tr>
<td>来</td>
<td>B-v</td>
<td>B-NN</td>
<td>来访</td>
<td>B-v</td>
<td>B-NN</td>
</tr>
<tr>
<td>访</td>
<td>I-v</td>
<td>I-NN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
An example

Example

刘 B-nr B-NR 刘 B-nr B-NR
华 B-nr I-NR 华清 B-nr I-NR
清 I-nr I-NR
副 B-b B-NN 副 B-b B-NN
总 B-n I-NN 总理 B-n I-NN
理 I-n I-NN
的 B-u B-DEC 的 B-u B-DEC
这 B-r B-DT 这 B-r B-DT
次 I-r B-M 次 I-r B-M
来 B-v B-NN 来访 B-v B-NN
访 I-v I-NN
An example

Example

刘 B-nr B-NR 刘 B-nr B-NR
华 B-nr I-NR 华 B-nr I-NR
清 I-nr I-NR 清 B-nr I-NR
副 B-b B-NN 副 B-b B-NN
总 B-n I-NN 总理 B-n I-NN I-NN
理 I-n I-NN
的 B-u B-DEC 的 B-u B-DEC
这 B-r B-DT 这 B-r B-DT
次 I-r B-M 次 I-r B-M
来 B-v B-NN 来访 B-v B-NN
访 I-v I-NN
An example

Example
An example

Example

刘 B-nr B-NR 刘 B-nr B-NR B-NR 刘华清/NR
华 B-nr I-NR 华清 B-nr I-NR I-NR
清 I-nr I-NR 清 B-nr I-NR I-NR
副 B-b B-NN 副 B-b B-NN B-NN 副总理/NN
总 B-n I-NN 总理 B-n I-NN I-NN
理 I-n I-NN
的 B-u B-DEC 的 B-u B-DEC B-DEC 的/DEG
次 B-r B-DT 次 B-r B-DT B-DT 这/DT
来 I-r B-M 次 I-r B-M B-M 次/M
访 B-v B-NN 来访 B-v B-NN B-NN 来访/NN
Practical issues

**System architecture**

- Raw sentences
  - Heterogeneous solver A
  - Heterogeneous solver B
  - Heterogeneous solver C
  - Structured sentences
  - Structured sentences
  - Structured sentences
  - Merging
  - Sub-word sequences
  - Sub-word tagger SubTag

**Level 0 solvers**:
- Same data, different models (my 2011 paper)
- Different data, same model

Process target annotations to generate training data for the **level 1 solver**:
- heterogeneous level 0 solvers
- homogeneous level 0 solver: cross-validation/stacking
About training

$D_B \quad \Rightarrow \quad \text{Train Level 0 } B\text{-style tagger } T_B^0$
About training

\[ \mathcal{D}_B \Rightarrow \text{Train Level 0 } B\text{-style tagger } T^0_B \]

\[ \mathcal{D}_A \Rightarrow \text{Train Level 0 } A\text{-style tagger } T^0_A \]
About training

\[ D_B \quad \Rightarrow \quad \text{Train Level 0 } B\text{-style tagger } T_B^0 \]

\[ D_A \quad \Rightarrow \quad \text{Train Level 0 } A\text{-style tagger } T_A^0 \]

\[ D_A \Rightarrow \hat{B}(D_A) \quad \Rightarrow \quad \text{Label } D_A \text{ with } T_B^0 \]
About training

- \( \mathcal{D}_B \) \( \Rightarrow \) Train Level 0 \( B \)-style tagger \( T_B^0 \)

- \( \mathcal{D}_A \) \( \Rightarrow \) Train Level 0 \( A \)-style tagger \( T_A^0 \)

- \( \mathcal{D}_A \Rightarrow \hat{B}(\mathcal{D}_A) \) \( \Rightarrow \) Label \( \mathcal{D}_A \) with \( T_B^0 \)

- \( \mathcal{D}_A^{(1)} \), \( \mathcal{D}_A^{(2)} \), \( \mathcal{D}_A^{(3)} \) \( \Rightarrow \) Cross-validation
About training

- $\mathcal{D}_B$  ⇒  Train Level 0 $B$-style tagger $T^0_B$
- $\mathcal{D}_A$  ⇒  Train Level 0 $A$-style tagger $T^0_A$
- $\mathcal{D}_A \Rightarrow \hat{B}(\mathcal{D}_A)$  ⇒  Label $\mathcal{D}_A$ with $T^0_B$
- Test  Train  Train  ⇒  Get $\hat{A}(\mathcal{D}_A^{(1)})$
About training

$\mathcal{D}_B$ ⇒ Train Level 0 $B$-style tagger $T^0_B$

$\mathcal{D}_A$ ⇒ Train Level 0 $A$-style tagger $T^0_A$

$\mathcal{D}_A \Rightarrow \hat{B}(\mathcal{D}_A)$ ⇒ Label $\mathcal{D}_A$ with $T^0_B$

Train, Test, Train

⇒ Get $\hat{A}(\mathcal{D}_A^{(1)})$

Train, Test, Train

⇒ Get $\hat{A}(\mathcal{D}_A^{(2)})$
About training

$\mathcal{D}_B$ \implies \text{Train Level 0 } B\text{-style tagger } T^0_B$

$\mathcal{D}_A$ \implies \text{Train Level 0 } A\text{-style tagger } T^0_A$

$\mathcal{D}_A \Rightarrow \hat{B}(\mathcal{D}_A)$ \implies \text{Label } \mathcal{D}_A \text{ with } T^0_B$

$\begin{array}{ccc}
\text{Test} & \text{Train} & \text{Train} \\
\text{Train} & \text{Test} & \text{Train} \\
\text{Train} & \text{Train} & \text{Test}
\end{array}$ \implies \begin{align*}
\text{Get } \hat{A}(\mathcal{D}^{(1)}_A) \\
\text{Get } \hat{A}(\mathcal{D}^{(2)}_A) \\
\text{Get } \hat{A}(\mathcal{D}^{(3)}_A)
\end{align*}
## About training

<table>
<thead>
<tr>
<th>$\mathcal{D}_B$</th>
<th>$\Rightarrow$</th>
<th>Train Level 0 $B$-style tagger $T^0_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{D}_A$</td>
<td>$\Rightarrow$</td>
<td>Train Level 0 $A$-style tagger $T^0_A$</td>
</tr>
<tr>
<td>$\mathcal{D}_A \Rightarrow \hat{B}(\mathcal{D}_A)$</td>
<td>$\Rightarrow$</td>
<td>Label $\mathcal{D}_A$ with $T^0_B$</td>
</tr>
<tr>
<td>Test</td>
<td>Train</td>
<td>Train</td>
</tr>
<tr>
<td>Train</td>
<td>Test</td>
<td>Train</td>
</tr>
<tr>
<td>Train</td>
<td>Train</td>
<td>Test</td>
</tr>
<tr>
<td>$\mathcal{D}_A, \hat{A}(\mathcal{D}_A), \hat{B}(\mathcal{D}_A)$</td>
<td>$\Rightarrow$</td>
<td>Train Level 1 $A$-style sub-word tagger $T^1_A$</td>
</tr>
</tbody>
</table>
Outline

Annotation ensemble

Joint word segmentation and POS tagging

A sub-word tagging model

Experiments
Main results

We focus on improving CTB-style tagging with PPD.

<table>
<thead>
<tr>
<th>Test</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>State-of-the-art</td>
<td>94.02</td>
</tr>
<tr>
<td>Base model</td>
<td>93.41</td>
</tr>
<tr>
<td>+Re-training</td>
<td>94.11</td>
</tr>
<tr>
<td>Sub-word model</td>
<td>94.36</td>
</tr>
<tr>
<td>+Re-training</td>
<td>94.68</td>
</tr>
</tbody>
</table>

F-scores of different systems.
Main results

We focus on improving CTB-style tagging with PPD.

<table>
<thead>
<tr>
<th>Test</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>State-of-the-art</td>
<td>94.02</td>
</tr>
<tr>
<td>Base model</td>
<td>93.41</td>
</tr>
<tr>
<td>+ Re-training</td>
<td>94.11</td>
</tr>
<tr>
<td>Sub-word model</td>
<td>94.36</td>
</tr>
<tr>
<td>+ Re-training</td>
<td>94.68</td>
</tr>
</tbody>
</table>

Stacking model works! Approximation error is reduced!
Main results

We focus on improving CTB-style tagging with PPD.

<table>
<thead>
<tr>
<th>Test</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>State-of-the-art</td>
<td>94.02</td>
</tr>
<tr>
<td>Base model</td>
<td>93.41</td>
</tr>
<tr>
<td>+Re-training</td>
<td>94.11</td>
</tr>
<tr>
<td>Sub-word model</td>
<td>94.36</td>
</tr>
<tr>
<td>+Re-training</td>
<td>94.68</td>
</tr>
</tbody>
</table>

Corpus conversion works! Estimation error is reduced!
Main results

We focus on improving CTB-style tagging with PPD.

<table>
<thead>
<tr>
<th>Test</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>State-of-the-art</td>
<td>94.02</td>
</tr>
<tr>
<td>Base model</td>
<td>93.41</td>
</tr>
<tr>
<td>+Re-training</td>
<td>94.11</td>
</tr>
<tr>
<td>Sub-word model</td>
<td>94.36</td>
</tr>
<tr>
<td>+Re-training</td>
<td>94.68</td>
</tr>
</tbody>
</table>

Corpus conversion works! Estimation error is reduced!
Main results

We focus on improving CTB-style tagging with PPD.

<table>
<thead>
<tr>
<th>Test</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>State-of-the-art</td>
<td>94.02</td>
</tr>
<tr>
<td>Base model</td>
<td>93.41</td>
</tr>
<tr>
<td>+Re-training</td>
<td>94.11</td>
</tr>
<tr>
<td>Sub-word model</td>
<td>94.36</td>
</tr>
<tr>
<td>+Re-training</td>
<td>94.68</td>
</tr>
</tbody>
</table>

Better than previous results.
Inconsistent inputs:

• $CTag_{ppd \rightarrow ctb}$ and $STag_{ppd \rightarrow ctb}$ are trained on
  \[
  \{ \langle x, \hat{y}_{ppd}, *, y_{ctb} \rangle, \ldots \}.
  \]

• But when applying them for corpus conversion, we use gold $y_{ppd}$'s.
Answer consistency

Inconsistent inputs:

- \( \text{CTag}_{\text{ppd}\rightarrow\text{ctb}} \) and \( \text{STag}_{\text{ppd}\rightarrow\text{ctb}} \) are trained on \( \{ \langle x, \hat{y}_{\text{ppd}}, *, y_{\text{ctb}} \rangle, \ldots \} \).
- But when applying them for corpus conversion, we use gold \( y_{\text{ppd}} \)'s.

Stacking models trained with noisy inputs can \textit{tolerate} perfect inputs.

<table>
<thead>
<tr>
<th></th>
<th>Auto PPD</th>
<th>Gold PPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{CTag}_{\text{ppd}\rightarrow\text{ctb}} )</td>
<td>93.69</td>
<td>95.19</td>
</tr>
<tr>
<td>( \text{STag}_{\text{ppd}\rightarrow\text{ctb}} )</td>
<td>94.14</td>
<td>94.70</td>
</tr>
</tbody>
</table>

- \( \text{CTag}_{\text{ppd}\rightarrow\text{ctb}} \): Feature-based stacking model
- \( \text{STag}_{\text{ppd}\rightarrow\text{ctb}} \): Structure-based stacking model
Corpus conversion

- **CTag\textsubscript{ctb}**: Character-based baseline
- **\(D\textsubscript{ctb}\)**: CTB training data (original)
- **\(D\textsubscript{ppd}\)**: PPD training data (original)
- **\(D'\textsubscript{ctb} = D^{\text{CTag}_{ppd\rightarrow ctb}}\textsubscript{ppd\rightarrow ctb}\)**: Process \(D\textsubscript{ppd}\) with \(\text{CTag}_{ppd\rightarrow ctb}\) (converted)
- **\(D''\textsubscript{ctb} = D^{\text{STag}_{ppd\rightarrow ctb}}\textsubscript{ppd\rightarrow ctb}\)**: Process \(D\textsubscript{ppd}\) with \(\text{STag}_{ppd\rightarrow ctb}\) (converted)

<table>
<thead>
<tr>
<th></th>
<th>(\text{CTag}_{ctb})</th>
<th>(\text{STag}_{ppd\rightarrow ctb})</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D\textsubscript{ctb} \cup D'\textsubscript{ctb})</td>
<td>- -</td>
<td></td>
<td>94.26</td>
</tr>
<tr>
<td>(D\textsubscript{ctb} \cup D'\textsubscript{ctb})</td>
<td>(D\textsubscript{ctb})</td>
<td></td>
<td>94.52</td>
</tr>
<tr>
<td>(D\textsubscript{ctb})</td>
<td>(D\textsubscript{ctb} \cup D''\textsubscript{ctb})</td>
<td></td>
<td>94.06</td>
</tr>
<tr>
<td>(D\textsubscript{ctb} \cup D'\textsubscript{ctb})</td>
<td>(D\textsubscript{ctb} \cup D''\textsubscript{ctb})</td>
<td></td>
<td>94.62</td>
</tr>
</tbody>
</table>

Evaluation of retrained models.
Corpus conversion

- **CTag\textsubscript{ctb}**: Character-based baseline
- **D\textsubscript{ctb}**: CTB training data (original)
- **D\textsubscript{ppd}**: PPD training data (original)
- **D'\textsubscript{ctb} = D\textsubscript{ppd} → CTag\textsubscript{ppd} → ctb**: Process **D\textsubscript{ppd}** with **CTag\textsubscript{ppd} → ctb** (converted)
- **D''\textsubscript{ctb} = D\textsubscript{ppd} → STag\textsubscript{ppd} → ctb**: Process **D\textsubscript{ppd}** with **STag\textsubscript{ppd} → ctb** (converted)

<table>
<thead>
<tr>
<th>CTag\textsubscript{ctb}</th>
<th>STag\textsubscript{ppd} → ctb</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>D\textsubscript{ctb} \cup D'\textsubscript{ctb}</td>
<td>- -</td>
<td>94.26</td>
</tr>
<tr>
<td>D\textsubscript{ctb} \cup D'\textsubscript{ctb}</td>
<td>D\textsubscript{ctb}</td>
<td>94.52</td>
</tr>
<tr>
<td>D\textsubscript{ctb} \cup D''\textsubscript{ctb}</td>
<td>D\textsubscript{ctb} \cup D''\textsubscript{ctb}</td>
<td>94.06</td>
</tr>
<tr>
<td>D\textsubscript{ctb} \cup D'\textsubscript{ctb}</td>
<td>D\textsubscript{ctb} \cup D''\textsubscript{ctb}</td>
<td>94.62</td>
</tr>
</tbody>
</table>

Baseline=92.93! Corpus conversion works!
Corpus conversion

- \( \text{CTag}_{\text{ctb}} \): Character-based baseline
- \( D_{\text{ctb}} \): CTB training data (original)
- \( D_{\text{ppd}} \): PPD training data (original)
- \( D'_{\text{ctb}} = D_{\text{ppd}} \xrightarrow{\text{CTag}_{\text{ppd}} \rightarrow \text{ctb}} D_{\text{ctb}} \): Process \( D_{\text{ppd}} \) with \( \text{CTag}_{\text{ppd}} \rightarrow \text{ctb} \) (converted)
- \( D''_{\text{ctb}} = D_{\text{ppd}} \xrightarrow{\text{STag}_{\text{ppd}} \rightarrow \text{ctb}} D_{\text{ctb}} \): Process \( D_{\text{ppd}} \) with \( \text{STag}_{\text{ppd}} \rightarrow \text{ctb} \) (converted)

<table>
<thead>
<tr>
<th>( \text{CTag}_{\text{ctb}} )</th>
<th>( \text{STag}_{\text{ppd} \rightarrow \text{ctb}} )</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D_{\text{ctb}} \cup D'_{\text{ctb}} )</td>
<td>- -</td>
<td>94.26</td>
</tr>
<tr>
<td>( D_{\text{ctb}} \cup D'_{\text{ctb}} )</td>
<td>( D_{\text{ctb}} )</td>
<td>94.52</td>
</tr>
<tr>
<td>( D_{\text{ctb}} \cup D''_{\text{ctb}} )</td>
<td>( D_{\text{ctb}} \cup D''_{\text{ctb}} )</td>
<td>94.06</td>
</tr>
<tr>
<td>( D_{\text{ctb}} \cup D'_{\text{ctb}} )</td>
<td>( D_{\text{ctb}} \cup D''_{\text{ctb}} )</td>
<td>94.62</td>
</tr>
</tbody>
</table>

Baseline=94.03! Corpus conversion works!
Corpus conversion

- $CTag_{ctb}$: Character-based baseline
- $D_{ctb}$: CTB training data (original)
- $D_{ppd}$: PPD training data (original)
- $D'_{ctb} = D_{ppd}^{CTag_{ppd} \rightarrow_{ctb}}$: Process $D_{ppd}$ with $CTag_{ppd} \rightarrow_{ctb}$ (converted)
- $D''_{ctb} = D_{ppd}^{STag_{ppd} \rightarrow_{ctb}}$: Process $D_{ppd}$ with $STag_{ppd} \rightarrow_{ctb}$ (converted)

<table>
<thead>
<tr>
<th>$CTag_{ctb}$</th>
<th>$STag_{ppd} \rightarrow_{ctb}$</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{ctb} \cup D'_{ctb}$</td>
<td>- -</td>
<td>94.26</td>
</tr>
<tr>
<td>$D_{ctb} \cup D'_{ctb}$</td>
<td>$D_{ctb}$</td>
<td>94.52</td>
</tr>
<tr>
<td>$D_{ctb}$</td>
<td>$D_{ctb} \cup D''_{ctb}$</td>
<td>94.06</td>
</tr>
<tr>
<td>$D_{ctb} \cup D'_{ctb}$</td>
<td>$D_{ctb} \cup D''_{ctb}$</td>
<td>94.62</td>
</tr>
</tbody>
</table>

Feature-based stacking is more effective for corpus conversion.
Learning curves

<table>
<thead>
<tr>
<th>#CTB</th>
<th>#PPD</th>
<th>$\text{CTag}_{\text{ppd} \rightarrow \text{ctb}}$</th>
<th>$\text{STag}_{\text{ppd} \rightarrow \text{ctb}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>18104</td>
<td>7381</td>
<td>92.21</td>
<td>93.26</td>
</tr>
<tr>
<td>18104</td>
<td>14545</td>
<td>93.22</td>
<td>93.82</td>
</tr>
<tr>
<td>18104</td>
<td>21745</td>
<td>93.58</td>
<td>93.96</td>
</tr>
<tr>
<td>18104</td>
<td>28767</td>
<td>93.55</td>
<td>93.87</td>
</tr>
<tr>
<td>18104</td>
<td>35996</td>
<td>93.67</td>
<td>94.03</td>
</tr>
<tr>
<td>9052</td>
<td>9052</td>
<td>92.10</td>
<td>92.40</td>
</tr>
</tbody>
</table>

Learning curves.
### Learning curves

<table>
<thead>
<tr>
<th>#CTB</th>
<th>#PPD</th>
<th>CTag&lt;sub&gt;ppd→ctb&lt;/sub&gt;</th>
<th>STag&lt;sub&gt;ppd→ctb&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>18104</td>
<td>7381</td>
<td>92.21</td>
<td>93.26</td>
</tr>
<tr>
<td>18104</td>
<td>14545</td>
<td>93.22</td>
<td>93.82</td>
</tr>
<tr>
<td>18104</td>
<td>21745</td>
<td>93.58</td>
<td>93.96</td>
</tr>
<tr>
<td>18104</td>
<td>28767</td>
<td>93.55</td>
<td>93.87</td>
</tr>
<tr>
<td>18104</td>
<td>35996</td>
<td>93.67</td>
<td>94.03</td>
</tr>
<tr>
<td>9052</td>
<td>9052</td>
<td>92.10</td>
<td>92.40</td>
</tr>
</tbody>
</table>

Baseline=92.93! Feature-based stacking fails!
Learning curves

<table>
<thead>
<tr>
<th>#CTB</th>
<th>#PPD</th>
<th>C$\text{Tag}_{\text{ppd}\rightarrow\text{ctb}}$</th>
<th>S$\text{Tag}_{\text{ppd}\rightarrow\text{ctb}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>18104</td>
<td>7381</td>
<td>92.21</td>
<td>93.26</td>
</tr>
<tr>
<td>18104</td>
<td>14545</td>
<td>93.22</td>
<td>93.82</td>
</tr>
<tr>
<td>18104</td>
<td>21745</td>
<td>93.58</td>
<td>93.96</td>
</tr>
<tr>
<td>18104</td>
<td>28767</td>
<td>93.55</td>
<td>93.87</td>
</tr>
<tr>
<td>18104</td>
<td>35996</td>
<td>93.67</td>
<td>94.03</td>
</tr>
<tr>
<td>9052</td>
<td>9052</td>
<td>92.10</td>
<td>92.40</td>
</tr>
</tbody>
</table>

Structure-based stacking is robust!
## Learning curves

<table>
<thead>
<tr>
<th>#CTB</th>
<th>#PPD</th>
<th>CTag_{ppd→ctb}</th>
<th>STag_{ppd→ctb}</th>
</tr>
</thead>
<tbody>
<tr>
<td>18104</td>
<td>7381</td>
<td>92.21</td>
<td>93.26</td>
</tr>
<tr>
<td>18104</td>
<td>14545</td>
<td>93.22</td>
<td>93.82</td>
</tr>
<tr>
<td>18104</td>
<td>21745</td>
<td>93.58</td>
<td>93.96</td>
</tr>
<tr>
<td>18104</td>
<td>28767</td>
<td>93.55</td>
<td>93.87</td>
</tr>
<tr>
<td>18104</td>
<td>35996</td>
<td>93.67</td>
<td>94.03</td>
</tr>
<tr>
<td>9052</td>
<td>9052</td>
<td>92.10</td>
<td>92.40</td>
</tr>
</tbody>
</table>

Baseline=92.93!  We are not that lucky!
Conclusion and future work

Annotation ensemble is beneficial.
Conclusion and future work

Annotation ensemble is beneficial.

It is easy to extend this work to solve other annotation ensemble problems.
Conclusion and future work

Annotation ensemble is beneficial.

It is easy to extend this work to solve other annotation ensemble problems.

- The analysis of the possible impact of heterogeneous annotations is general.
Conclusion and future work

Annotation ensemble is beneficial.

It is easy to extend this work to solve other annotation ensemble problems.

• The analysis of the possible impact of heterogeneous annotations is general.
• The idea to leverage heterogeneous annotations to reduce approximation and estimation errors is general.
Conclusion and future work

Annotation ensemble is beneficial.

It is easy to extend this work to solve other annotation ensemble problems.

- The analysis of the possible impact of heterogeneous annotations is general.
- The idea to leverage heterogeneous annotations to reduce approximation and estimation errors is general.
- The stacking-based framework is general.
Game over
Game over

QUESTIONS?

COMMENTS?