A Machine Learning Approach to Recipe Flow Construction

Shinsuke Mori, Tetsuro Sasada, Yoko Yamakata, Koichiro Yoshino

Kyoto University

2012/08/28
Table of Contents

Overview

Recipe Text Analysis

Evaluation

Conclusion
What is Recipe?

- Describing the procedures for a dish
  - submitted to the Web
  - mainly written by house chefs
- One of the successful web contents
  - search, visualization, ...

- Recipe Flow [Momouchi 80, Hamada 00]
Recipe as a Text for Natural Language Processing

- Containing general NLP problems
  - Word identification or segmentation (WS)
  - Named entity recognition (NER)
  - Syntactic analysis (SA)
  - Predicate-argument structure (PAS) analysis
  - etc.

- Simple compared with newspaper articles, etc.
  - Few modalities
  - Simple in tense and aspect
  - Mainly indicative or imperative mood
  - Only one person (Chef)
Overall Design

1. Recipe text analysis
   - State of the art in NLP area
   - Domain adaptation to recipe texts

2. Flow construction
   - Not rule-based (hopefully)
   - Graph-based approach

3. Match with movies
Recipe Text Analysis

Execute the following steps in this order

1. WS: Word segmentation (Including stemming)
   - Only required for languages without whitespace (ja, zh)
   - Some canonicalization required even for en, fr, ...

2. NER: Named entity recognition
   - Food, Tool, Duration, Quantity, State, Action by the chef or foods

3. SA: Syntactic analysis
   - Grammatical relationship among NEs

4. PAS: Predicate-argument structure analysis
   - Semantic relationship among NEs

Output

煮立て (obj.:水-400cc, で:鍋)
boil(obj.:water 400cc, by:pot)
Step 1. Word Segmentation (word identification)

- **Input:** a sentence
  水4 0 0 c cを鍋で煮立て、沸騰したら中華スープの素を加えてよく溶かす。
  (Heat 400 cc of water in a pot, and when it boils, add Chinese soup powder and dissolve it well.)

- **Output:** a word sequence
  水|4−0−0|c−c|を|鍋|で|煮-立-て|、|
  沸-騰|し|た-ら|中-華|スー-プ|の|素|を|
  加-え|て|よ-く|溶-か|す|。

where “|” and “−” mean existence and non-existence of a word boundary.

※ No dictionary form of inflectional words is needed because our standard divides them into the stem and the ending.
Pointwise WS (KyTea) [Neubig 11]

- Binary classification problem at each point between chars

\[
\begin{align*}
  x_{i-2} & \ x_{i-1} & \ x_i & \ x_{i+1} & \ x_{i+2} & \ x_{i+3} \\
\text{Text:} & \quad \text{鍋で煮立て、沸騰した} \\
  t_i: & \quad \text{Decision point}
\end{align*}
\]

- Trainable from a partially annotated corpus
  ⇒ Flexible corpus annotation!
  ⇒ Easy to adapt to a specific domain!

- A partially annotated corpus allows us to focus on special terms
  弱火で煮-立-て|る
  これが煮-立|つ|まで
Pointwise WS (KyTea) [Neubig 11]

- Binary classification problem at each point between chars

\[ x_{i-2} x_{i-1} x_i x_{i+1} x_{i+2} x_{i+3} \]

Text: 鍋で煮立て、沸騰した

\[ t_i: \text{Decision point} \]

- SVM (Support Vector Machine)

- Features
  - Char (type) 1-gram feature:
    - -3/鍗 (K), -2/で (H), -1/煮 (K), 1/立て (K), 2/て (H), 3/、(S)
  - Char (type) 2-gram feature:
    - -3/鍗で (KH), -2/で煮 (HK), -1/立て (KK), 1/立て (KH), 2/て、(HS)
  - Char (type) 3-gram feature:
    - -3/鍗で煮 (KHK), -2/で煮立て (HKK), -1/煮立て (KKH), 1/立て、(KHS)
Baseline and its Adaptation

- **Baseline:** BCCWJ, UniDic, etc.
- **Adaptation:** KWIC based partial annotation
  - 8 hours
F measure = \{\left(\text{LCS/sysout}^{-1} + \text{LCS/corpus}^{-1}\right)/2\}^{-1}

- WS improves as the work time increases
- More work required (about 98% in the general domain)
Step 2. Named Entity Recognition (NER)

- Named entity
  - Word sequences corresponding to objects and actions in the real world
  - Highly domain dependent

- Named entity types for recipes:
  \textbf{Food}, \textbf{Tool}, \textbf{Duration}, \textbf{Quantity}, \textbf{State}, \textbf{Action} by the \textbf{chef}, \textbf{Action} by \textbf{foods}

\begin{quote}
水 F 4 0 0 c c Q を 鍋 T で 煮立て Ac て 沸騰し Af たら 中華スープの素 F を 加え Ac て よく溶か Ac ます。

Heat Ac 400 cc Q of water F in a pot T, and when it boils Af, add Chinese soup powder F and dissolve Ac it well.
\end{quote}
Pointwise NER

Trainable from a partially annotated corpus
⇒ Flexible corpus annotation!
⇒ Easy to adapt to a specific domain!

1. BIO2 representation (one NE tag for a word, with Other)
   水/B-F 4 O O/B-Q c c/I-Q を/O 鍋/BT で/O
   煮立て/B-Ac 、/O 沸騰/B-Af し/I-Af たら/O

2. Train pointwise classifier (KyTea) with logistic regression
   from a tagged data including partially annotated corpus
   ▶ No partially annotated corpus this time
   ▶ Cf. A CRF requires a fully annotated sentences.
Pointwise NER (cont’d)

3. Output all the possible pairs of tag and probability to fill the Viterbi table:

| P(y|w) | 水 | 4 | 0 | 0 | c | c | も | … |
|--------|----|---|---|---|---|---|---|---|
| F-B    | 0.62 | 0.00 | 0.00 | 0.00 | … |
| F-I    | 0.37 | 0.00 | 0.00 | 0.00 | … |
| Q-B    | 0.00 | 0.82 | 0.01 | 0.00 | … |
| y      | 0.00 | 0.17 | 0.99 | 0.00 | … |
| Q-I    | 0.00 | 0.00 | 0.00 | 0.00 | … |

4. Search for the best sequence satisfying the constraints
   - Ex. “F-I Q-I” is invalid
   - In future work we change this part into CRFs
Baseline and its Adaptation

- Baseline: 1/10 of *Meet-potato* recipe text (24 sent.)
- Annotation: from 1/10 to 10/10 (about 5 hours, 242 sent.)

*Not randomly selected recipes ... (bad setting)*

Meet potato
Result

- **F measure**

Very low F measure compared with the general domain (around 80%)

- **NER improves rapidly as the work time increases**
Step 3. Syntactic Analysis

- Dependency among the words (and NEs) in a sentence
Pointwise SA

- Pointwise MST (EDA) [Flannery 11]

Trainable from a partially annotated corpus
⇒ Flexible corpus annotation!
⇒ Easy to adapt to a specific domain!

1. Estimate dependency scores of all the possible pairs in a sentence

\[ \sigma(\langle i, d_i \rangle, \vec{w}), \quad \text{where } w_i \text{ depends on } w_{d_i} \]

2. Select the Spanning Tree which Maximizes the total score (MST)

\[ \hat{d} = \arg \max_{d \in D} \sum_{i=1}^{n} \sigma(\langle i, d_i \rangle, \vec{w}) \]
Pointwise SA (cont’d)

- Features for dependency score of a word pair

\[ \text{oyster} \quad \text{obj.} \quad \text{Hiroshima} \quad \text{to} \quad \text{eat} \quad \text{to} \quad \text{go} \quad \text{infl.} \]

\( w_{i-3} \quad w_{i-2} \quad w_{i-1} \quad w_i \quad w_{i+1} \quad w_{i+2} \quad w_{i+3} \)

\( w_{d_i-3} \quad w_{d_i-2} \quad w_{d_i-1} \quad w_{d_i} \quad w_{d_i+1} \quad w_{d_i+2} \quad w_{d_i+3} \)

F1  The distance between a dependent word \( w_i \) and its candidate head \( w_{d_i} \).
F2  The surface forms of \( w_i \) and \( w_{d_i} \).
F3  The parts-of-speech of \( w_i \) and \( w_{d_i} \).
F4  The surface forms of up to three words to the left of \( w_i \) and \( w_{d_i} \).
F5  The surface forms of up to three words to the right of \( w_i \) and \( w_{d_i} \).
F6  The parts-of-speech of the words selected for F4.
F7  The parts-of-speech of the words selected for F5.
Baseline and its Adaptation

- **Baseline:** about 20k sent.
  - EHJ (Dictionary example sentences): 11,700 sentences, 145,925 words
  - NKN (*Nikkei* newspaper articles): 9,023 sentences, 263,425 words

- **Adaptation:** Annotate new pairs of a noun and a postposition with the dependency
  1. Find a pair of a noun and a postposition not appearing in the training corpus
  2. Annotate the dependencies from the noun to its head verb
    \[ c \rightarrow \text{obj.} \rightarrow ( \cdots \text{ 煮立て} ) \]
  3. 8 hours
Result

- Accuracy

- Low accuracy compared with the in-domain data (96.83%)

- SA improves slowly as the work time increases
Step 4. Predicate-argument structure analysis

- Rule-based (as far as it is)
  - Should be based on a machine learning
  - Have to guess zero-pronouns

- Correspond to the smallest units in the recipe flow

1. \(\text{煮立て}_\text{Ac} (\text{Chef, 水}_F 400 \text{ cc} \text{ obj.} \rightarrow \text{pot}_T \text{ in})\)

400 cc of water (obj.)

2. \(\text{沸騰-し}_\text{Af} (\text{Food})\)

3. \(\text{加え}_\text{Ac} (\text{Chef, 中華スープの素}_F \text{ obj.} \rightarrow \text{水}_F \text{ in})\)

Chinese soup powder

4. \(\text{溶か-す}_\text{Ac} (\text{Chef, 中華スープの素}_F \text{ を})\)

dissolve
Experimental Setting

1. Test data: randomly selected 100 recipes in Japanese

<table>
<thead>
<tr>
<th></th>
<th>#recipes</th>
<th>#sent.</th>
<th>#words</th>
<th>#NEs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>724</td>
<td>13,150</td>
<td>3,797</td>
</tr>
</tbody>
</table>

2. Training data
   - WS: (BCCWJ + etc.) + partial annotation
   - NER: Meet-potato 1/10 + 9/10 (bad setting ...)
   - SA: (EHJ + NKN) + partial annotation
   - PAS: on going
   - Recipe Flow: on going
Evaluation 1: Each Step (summary)

Step 1. WS: Word segmentation
Baseline: 95.46%
⇓ (8 hours)
Adaptation: 95.84%

Step 2. NER: Named entity recognition
Baseline: 53.42%
⇓ (5 hours)
Annotation: 67.02%

Step 3. SA: Syntactic analysis
Baseline: 92.58%
⇓ (8 hours)
Adaptation: 93.02%
Evaluation 2: Overall

1. Predicate-argument structure
   - PA pair as an evaluation unit
     - \( \langle \text{煮立て, obj.:水-400 cc of water} \rangle \)
     - \( \langle \text{煮立て, で:鍋} \rangle \)
   - F measure
     - Baseline 42.01%
     - \( \downarrow (8 + 5 + 8 \text{ hours}) \) 28.0% error elimination!
     - Adaptation 58.27%
   - F measure is still low
     - Because of NER? (67.02% \( \ll 90\% \))
     - More annotation required (21 hours \( \ll \infty \))
     - Strict criterion (word boundary incl., etc.)

2. Flow Accuracy
Conclusion

- Recipe Text Analysis
  - Word segmentation, Named entity recognition
  - Syntactic analysis, Predicate-argument structure analysis

- A Machine Learning Approach
  - Systematic domain adaptation
  - Easily trainable to achieve the required accuracy

- Future work
  - Improvement
  - Recipe flow construction (search, visualization, ...)
  - Matching with movies to understand the real world
  - Spoken dialog system to help a chef (Smart kitchen)
    ▶ equipped with the recipe flow as the database
PNAT: Pointwise NLP Annotation Tool

- Word segmentation
- Part-of-speech tag
- Pronunciation
- Named entity tag
- Syntactic structure
References

