Improving Tokenization, Transcription Normalization and Part-of-speech Tagging of Ainu Language through Merging Multiple Dictionaries

Karol Nowakowski  Michal Ptaszynski  Fumito Masui
The Ainu people

* Native inhabitants of Hokkaidō.

* Estimated size of Ainu population in Hokkaidō – around 16 thousand people (Hokkaidō regional government, 2013).

Image source: https://commons.wikimedia.org
Ainu language

* Language isolate (no confirmed relation to any other language)

* SOV (Subject-Object-Verb) word order (same as Japanese)

* Polysynthetic (especially classical language, such as in yukar stories)

Example:

Iramante oruspe ka aeukoisoytak

Meaning: “We can also talk about hunting”

Source:
Current situation

* Only 7.2% of Ainu people are able to communicate in the Ainu language (survey by Hokkaido regional government conducted in 2013, with 586 respondents)

* Status: Critically endangered / nearly extinct

Example:

Iramante oruspe ka aeukoisoytak

Meaning: “We can also talk about hunting”

Ainu language preservation and revitalisation:

* Ainu language classes

* radio course (STV Radio, Sapporo)

* annual Ainu language speech contest (held by The Foundation for Research and Promotion of Ainu Culture),

* “The Ainu Times” (published quarterly)

* music groups singing in the Ainu language ("Oki", "Dub Ainu Band")

http://www.tonkori.com
Aims of this research

• create language analysis toolkit for the Ainu language
• facilitate analysis of the Ainu language by linguists and researchers of the Ainu literature
• contribute to the process of preservation and reviving of the Ainu language
Previous work – POST-AL

• In 2012 Ptaszynski and Momouchi created POST-AL (“Part of Speech Tagger for the Ainu Language).

• POST-AL performs the following tasks:

1. Transcription normalization – modification of parts of text that do not conform to modern rules of transcription (e.g. kamui -> kamuy).

Example:

<table>
<thead>
<tr>
<th>Original text:</th>
<th>Shineantota petetok un shinotash kushu payeash awa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized</td>
<td>Sineantota petetok un sinotas kusu payeas</td>
</tr>
<tr>
<td>transcription:</td>
<td>awa</td>
</tr>
<tr>
<td>Meaning:</td>
<td>“One day when I went for a trip up the river”</td>
</tr>
</tbody>
</table>
Previous work – POST-AL

- In 2012 Ptaszynski and Momouchi created POST-AL ("Part of Speech Tagger for the Ainu Language").
- POST-AL performs the following tasks:
  1. Transcription normalization – modification of parts of text that do not conform to modern rules of transcription (e.g. \textit{kamui} -> \textit{kamuy}).
  2. Word segmentation (tokenization) – a process in which the text is separated into tokens (words, punctuation marks, etc.), which become the basic unit for further analysis.

<table>
<thead>
<tr>
<th>Original text:</th>
<th>unnukar awa kor wenpuri enantui ka</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST-AL output (tokens):</td>
<td>un nukar a wa kor wen puri enan tuyka</td>
</tr>
<tr>
<td>Token 1</td>
<td>Token 2</td>
</tr>
<tr>
<td>Meaning:</td>
<td>“When she found me, her face [took] the color of anger.”</td>
</tr>
</tbody>
</table>
Previous work – POST-AL

- In 2012 Ptaszynski and Momouchi created POST-AL ("Part of Speech Tagger for the Ainu Language).

POST-AL performs the following tasks:

1. Transcription normalization – modification of parts of text that do not conform to modern rules of transcription (e.g. kamui -> kamuy).
2. Word segmentation (tokenization) – a process in which the text is separated into tokens (words, punctuation marks, etc.), which become the basic unit for further analysis.
3. Part-of-speech tagging – assigning a part-of-speech marker to each token.

Example:

POST-AL tagger iyosno ku hosipire kusne na
output: 【副】【人接】【他】【助動】【終助】
Meaning: “I'll return it later”
Previous work – POST-AL

• In 2012 Ptaszynski and Momouchi created POST-AL (“Part of Speech Tagger for the Ainu Language).

• POST-AL performs the following tasks:

1. Transcription normalization – modification of parts of text that do not conform to modern rules of transcription (e.g. kamui -> kamuy).

2. Word segmentation (tokenization) – a process in which the text is separated into tokens (words, punctuation marks, etc.), which become the basic unit for further analysis.

3. Part-of-speech tagging – assigning a part-of-speech marker to each token.


Example:

POST-AL tagger iyosno ku hosipire kusne na
output:【副】【人接】【他】【助動】【終助】
最後に、終わり、後から、後で 私は、私が、私の 返す つもりである よ、か

Meaning: “I'll return it later"
POST-AL’s dictionary base


• 2,019 entries

• The dictionary has been transformed to XML format

• Each entry contains:
  1. Token (word, morpheme, etc.)
  2. Part of speech
  4. Usage examples (not for all entries)
  5. Reference to yukar story it appears in (not for all entries)

Sample entry:

```xml
<word>aep</word>
<morph>a$^{2}$-e$^{1}$-p$^{1}$</morph>
<pos>名詞</pos>
<tr>食べ物</tr>
<ref>aep'omuken</ref>
```
Improving transcription normalization

Transcription change rules:

<table>
<thead>
<tr>
<th>Original transcription</th>
<th>Modern transcription standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch sh(i) ai ui ei oi au iu eu ou b g d m</td>
<td>chepshuttuye chiki</td>
</tr>
<tr>
<td>c s ay uy ey oy aw iw ew ow p k t n</td>
<td>chepshuttuye chiki</td>
</tr>
</tbody>
</table>

Input: chepshuttuye chiki  
Processing: chepshuttuye chiki  
Output: chepshuttuye chiki 
Transcription changes are optional
Improving transcription normalization

Transcription change rules:

<table>
<thead>
<tr>
<th>Original transcription</th>
<th>Modern transcription standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch</td>
<td>c</td>
</tr>
<tr>
<td>sh(i)</td>
<td>s</td>
</tr>
<tr>
<td>ai</td>
<td>ay</td>
</tr>
<tr>
<td>ui</td>
<td>uy</td>
</tr>
<tr>
<td>ei</td>
<td>ey</td>
</tr>
<tr>
<td>oi</td>
<td>oy</td>
</tr>
<tr>
<td>au</td>
<td>aw</td>
</tr>
<tr>
<td>iu</td>
<td>iw</td>
</tr>
<tr>
<td>eu</td>
<td>ew</td>
</tr>
<tr>
<td>ou</td>
<td>ow</td>
</tr>
<tr>
<td>b</td>
<td>p</td>
</tr>
<tr>
<td>g</td>
<td>k</td>
</tr>
<tr>
<td>d</td>
<td>t</td>
</tr>
<tr>
<td>m</td>
<td>n</td>
</tr>
</tbody>
</table>

Input: setautar

Processing: setautar

Output: setawtar

Morpheme boundary (seta-utar – “dogs”)
Improving tokenizer

Input string:

List of all matching words found in the dictionary base:

Possible tokenizations:

Output:

Tokenizer stops after finding the first possible match (which has the smallest number of tokens)
Improving tokenizer

PROBLEM: This tokenization algorithm always prefers long words over shorter ones.

Input string:
chiki
ciki

List of all matching words found in the dictionary base:
ciki
cik
cik
iki
ci
ki
i

Possible tokenizations:
1 TOKEN: ciki
ciki

2 TOKENS: cik i
ck i

Output:
ciki

CORRECT TOKENIZATION
Improving part-of-speech tagger

“Tagging is a disambiguation task” (some words have more than one possible part-of-speech) (Jurafsky and Martin, 2016. *Speech and Language Processing*)

Input: .kekorka

Matching entries found in the dictionary base:

1. `<word>ki</word>` <pos>二項動詞</pos>...
2. `<word>ki</word>` <pos>助動詞</pos>...
3. `<word>ki</word>` <pos>位置名詞長形形成接尾辞</pos>...

Output:  ???

Two methods of POS disambiguation applied in POST-AL:
1. N-gram based POS disambiguation
2. Term Frequency (TF) based POS disambiguation
Improving part-of-speech tagger

N-gram based POS disambiguation:
* Uses sample sentences included in the dictionary base for determining the correct POS tag

Checks word n-grams (trigrams) instead of just single words.

Input: kɪp ne⁰⁰

Matching entries found in the dictionary base:

1. <word>ki</word>
   <pos>二項動詞</pos>
   <ex>inkar he tap nep
tap teta ki humi okay</ex>
   <ex>... newa ci kɪp ne⁰⁰
korka</ex>
   <ex>ki a ine no</ex>
   <ex>kɪp ne⁰⁰korka</ex>

2. <word>ki</word>
   <pos>助動詞</pos>
   <ex>he ki</ex>
   <ex>ki humi okay</ex>
   <ex>ki kuni ne</ex>
   <ex>ki kusne</ex>
   <ex>ki rok okay</ex>
   <ex>ki ruwe ne</ex>
   <ex>ki ruwe okay</ex>
   <ex>ki siri ne</ex>
   <ex>ki siri tap an</ex>
   <ex>ki wa</ex>
   <ex>ki wa kusu</ex>
   <ex>ki wa ne yakka</ex>
   <ex>ki ya</ex>
   <ex>sir an ki ko</ex>

3. <word>ki</word>
   <pos>位置名詞長形形成接尾辞</pos>
Improving part-of-speech tagger

TF based POS disambiguation:

* Checks term frequency of each candidate word (= number of sample sentences included in the dictionary base) for determining the correct POS tag

Input: kip ne korka

Matching entries found in the dictionary base:

1. 1
   <word>ki</word>
   <pos>二項動詞</pos>
   <ex>inkar he tap nep tap teta ki humi okay</ex>
   <ex>... ne... ci ki p ne korka</ex>
   <ex>ki a ine no</ex>
   <ex>ki p ne korka</ex>

2. 14
   <word>ki</word>
   <pos>助動詞</pos>
   <ex>he ki</ex>
   <ex>ki humi okay</ex>
   <ex>ki kuni ne</ex>
   <ex>ki kusne</ex>
   <ex>ki rok okay</ex>
   <ex>ki ru ne</ex>
   <ex>ki ruw okay</ex>
   <ex>ki siri ne</ex>
   <ex>ki siri tap an</ex>
   <ex>ki wa</ex>
   <ex>ki wa kusu</ex>
   <ex>ki wa ne yakka</ex>
   <ex>ki ya</ex>
   <ex>sir an ki ko</ex>

3. 0
   <word>ki</word>
   <pos>位置名詞長形形成接尾辞</pos>
Improving part-of-speech tagger

• Word n-grams are more reliable as a method for POS disambiguation

• On the other hand, for many cases there are no relevant usage examples in the dictionary base

• To compensate for that, we created a modified tagging algorithm, which in such cases also takes into account the Term Frequency
Expanding POST-AL’s dictionary base

Dictionaries used:


Sample entry:

<word>aep</word>
<morph>a$^\{2\}$-e$^\{1\}$-p$^\{1\}$</morph>
<pos>名詞</pos>
<tr>食べ物</tr>
<ref>aep'omuka</ref>
Expanding POST-AL’s dictionary base

Dictionaries used:

Expanding POST-AL's dictionary base

Dictionaries used:


Sample entry (original):

此村に何か食物があるか
Tan kotan ta nepka aep an ruwe he an?

Is there anything to eat in this village?

Original entries often consist of more than one word (multiple words or phrases)
Expanding POST-AL’s dictionary base

Dictionaries used:


Sample entry (original):

```
此村に何か食べ物があるか
Tan kotan ta nepka aep an ruwe he an?
tan kotan ta nepka aep an ruwe an?
タン コタン タ ネプ カ アエプ アン ルウェ アン?
この村に何か食べ物あることある
【連体】【名】【格助】【疑問】【副助】【名】【自】【形名】【自】
「この村に何か食べ物ありますか？」
“Is there anything to eat in this village?”
```

Sample entry (modified dictionary):

```
<word>aep</word><kana>アエプ</kana>
<morph>a-e-p</morph><pos>【名】</pos>
<pos_en>n</pos_en>
<tr>食べ物</tr><tr_en>food</tr_en>
<ge>INDF.A-eat-thing</ge>
<ex>tan kotan ta nep ka aep an ruwe an?</ex>
<ex_jp>この村に何か食べ物はありますか？</ex_jp>
<ex_en>Is there anything to eat in this village?</ex_en>
```
Expanding POST-AL’s dictionary base

Dictionaries used:
3. Combined dictionary (1+2).
A) extracted entries containing words listed in both dictionaries
B) automatically unified duplicate entries, basing on their Japanese translations (at least one kanji character in common)

Entry from *Ainu shin-yōshū jiten*

```
<word>aep</word>
<morph>a$^2$-e$^1$-p$^1$</morph>
<pos>名詞</pos>
<tr>食べ物</tr>
<ref>aep'omuken</ref>
```

Entry from *Ainu Conversational Dictionary*

```
<word>aep</word><kana>アエプ</kana><morph>a-e-p</morph><pos>【名】</pos>
<pos_en>n</pos_en><tr>食べ物</tr><tr_en>food</tr_en><ge>INDF.A-eat-thing</ge><ex>tan kotan ta nep ka aep an ruwe an?</ex><ex_jp>この村に何か食べ物はありますか？</ex_jp><ex_en>Is there anything to eat in this village?</ex_en>
```
Expanding POST-AL’s dictionary base

Dictionaries used:
3. Combined dictionary (1+2).
A) extracted entries containing words listed in both dictionaries
B) automatically unified duplicate entries, basing on their Japanese translations (at least one kanji character in common)
C) that resulted in a dictionary containing 4,161 entries.

```
<word>aep</word><kana>アエプ</kana>
<morph_kk>\text{a}^{2}\text{e}^{1}\text{p}^{1}</morph_kk>
<morph_jk>a-e-p</morph_jk>
<pos_jk>【名】</pos_jk>
<pos_kk>名詞</pos_kk>
<pos_en>n</pos_en>
<tr>食べ物</tr><tr_en>food</tr_en>
<ex>tan kotan ta nep ka aep an ruwe an?</ex>
<ex_jp>この村に何か食べ物はありますか？</ex_jp>
<ex_en>Is there anything to eat in this village?</ex_en>
<ge>INDF.A-eat-thing</ge>
<ref>aep'omuken</ref>
```
Entry from combined dictionary
### Evaluation experiments

#### Transcription normalization results:

<table>
<thead>
<tr>
<th>DICTIONARY</th>
<th>Avg. result (F-score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Ainu shin-yōshū jiten (Kirikae)</td>
<td>91.85%</td>
</tr>
<tr>
<td>2. Ainu Conversational Dictionary (Jinbō and Kanazawa)</td>
<td>87.96%</td>
</tr>
<tr>
<td>3. Combined dictionary (1+2)</td>
<td><strong>92.48%</strong></td>
</tr>
</tbody>
</table>

#### Tokenization results:

<table>
<thead>
<tr>
<th>DICTIONARY</th>
<th>Avg. result (F-score)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Ainu shin-yōshū jiten (Kirikae)</td>
<td><strong>86.73%</strong></td>
</tr>
<tr>
<td>2. Ainu Conversational Dictionary (Jinbō and Kanazawa)</td>
<td>69.93%</td>
</tr>
<tr>
<td>3. Combined dictionary (1+2)</td>
<td><strong>87.73%</strong></td>
</tr>
</tbody>
</table>
## Evaluation experiments

### POS tagging results:

<table>
<thead>
<tr>
<th>DICTIOARY</th>
<th>Avg. result (F-score)</th>
<th>Tagging algorithm version:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg.</td>
<td>N-grams</td>
</tr>
<tr>
<td>1. <em>Ainu shin-yōshū jiten</em> (Kirikae)</td>
<td>72.16%</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>71.71%</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>74.72%</td>
<td>YES</td>
</tr>
<tr>
<td>2. <em>Ainu Conversational Dictionary</em> (Jinbō and Kanazawa)</td>
<td>80.01%</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>77.28%</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>81.55%</td>
<td>YES</td>
</tr>
<tr>
<td>3. Combined dictionary (1+2)</td>
<td>90.62%</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>90.27%</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td><strong>92.82%</strong></td>
<td>YES</td>
</tr>
</tbody>
</table>
Conclusions

1. Improved the following functions of POST-AL:
   • Transcription normalization
   • Tokenizer
   • POS tagger

2. Expanded POST-AL’s dictionary base by combining 2 dictionaries:
   • found out that the combination improved overall performance of the system
Thank you for your attention!
Evaluation experiments

Applied datasets:

• Yukar (9-13) from Ainu shin-yōshū ("Ainu Songs of Gods")
• JK dictionary sample sentences
• Sample text from Masayoshi Shibatani’s The Languages of Japan
• Mukawa dialect samples (by Chiba University)
Evaluation experiments

Statistics of unknown words:

<table>
<thead>
<tr>
<th></th>
<th>TEST DATA</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yukar 9-13</td>
<td>JK samples</td>
<td>Shib.</td>
<td>Muk.</td>
</tr>
<tr>
<td>WORDS TOTAL</td>
<td>1613</td>
<td>428</td>
<td>154</td>
<td>87</td>
</tr>
<tr>
<td>JK</td>
<td>431</td>
<td>0</td>
<td>32</td>
<td>11</td>
</tr>
<tr>
<td>JK+KK</td>
<td>14</td>
<td>0</td>
<td>23</td>
<td>10</td>
</tr>
<tr>
<td>KK</td>
<td>15</td>
<td>84</td>
<td>48</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>26,72%</td>
<td>0,00%</td>
<td>20,78%</td>
<td>12,64%</td>
</tr>
<tr>
<td></td>
<td>0,93%</td>
<td>19,63%</td>
<td>31,17%</td>
<td>22,99%</td>
</tr>
<tr>
<td></td>
<td>0,87%</td>
<td>0,00%</td>
<td>14,94%</td>
<td>11,49%</td>
</tr>
</tbody>
</table>
## Evaluation experiments

### Transcription normalization experiment results (F-score):

<table>
<thead>
<tr>
<th></th>
<th>TEST DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DICTIONARY</td>
<td></td>
</tr>
<tr>
<td>JK</td>
<td>92.34% 91.35%</td>
</tr>
<tr>
<td>KK</td>
<td>97.18% 98.55%</td>
</tr>
<tr>
<td>JK+KK</td>
<td>96.43% 97.11% 94.80% 91.41% 96.79% 78.32% 92.48%</td>
</tr>
</tbody>
</table>

Relatively low results for sample sentences from JK dictionary.

Explanation:
- We decided not to apply some of the transcription change rules observed only in that dictionary (such as ‘ra’→’r’ (e.g. arapa→arpa) or ‘ei’→’e’ (e.g. reihe→rehe)), as initial tests indicated that including them in the algorithm can cause errors with processing yukars and other texts.
## Evaluation experiments

### Tokenization experiment results (F-score):

<table>
<thead>
<tr>
<th>DICTIONARY</th>
<th>TEST DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yukar 9</td>
</tr>
<tr>
<td>JK</td>
<td>66.53%</td>
</tr>
<tr>
<td>KK</td>
<td>89.23%</td>
</tr>
<tr>
<td>JK+KK</td>
<td>85.37%</td>
</tr>
</tbody>
</table>
Evaluation experiments

<table>
<thead>
<tr>
<th>DICTIONARY</th>
<th>TEST DATA</th>
<th>Tagging algorithm version:</th>
<th>TEST DATA</th>
<th>Tagging algorithm version:</th>
</tr>
</thead>
<tbody>
<tr>
<td>JK</td>
<td>63.51%</td>
<td>YES</td>
<td>96.50%</td>
<td>NO</td>
</tr>
<tr>
<td>JK+KK</td>
<td>81.39%</td>
<td>YES</td>
<td>62.92%</td>
<td>NO</td>
</tr>
<tr>
<td>JK+KK</td>
<td>85.25%</td>
<td>YES</td>
<td>96.00%</td>
<td>NO</td>
</tr>
</tbody>
</table>

The gap between the results of tagging Yukar 10 and samples from JK dictionary can be partially explained by differences in part of speech classification of certain words between the two dictionaries applied in the system and the annotations (gold standard) provided by Momouchi (2008). For example, Momouchi annotated the word *ne* ("to be") as ‘auxiliary verb’, whereas in the dictionary base it is listed as ‘transitive verb’. 
Future tasks

1. Develop a tokenization algorithm based on word n-grams rather than single words.

2. Enlarge the dictionary base by adding other dictionaries, such as the *Ainu-Japanese Dictionary: Saru Dialect* by Suzuko Tamura.

3. Expand the dictionary base with the information about alternative transcription methods appearing in older texts (in order to improve the normalization of transcription in such texts).

4. Build a statistical model of the Ainu language, reflecting probability distribution over different sequences (bigrams or trigrams) of parts of speech, and use it to improve POS tagging performance.