Mostly-Unsupervised Statistical Segmentation of Japanese: Applications to Kanji

Rie Kubota and Lillian Lee

Cornell University

Japanese NLP

- Words/Characters are unspaced, so segmentation is an essential first step
- Current methods employ:
  - Pre-existing lexicon
  - Pre-existing grammar
  - Pre-segmented data
- English parallel: “theyouthevent”

Japanese Language

- 3 Types of Characters
  - kanji, hiragana, katakana
  - Are used within the same document, sentence, etc. (helps find <60% word boundaries)
  - The latter 2 often represent sounds (like English characters)

Kanji

- Are often:
  - Domain terms or Proper nouns (unknown word problem, important for IR)
  - Compound nouns (POS doesn’t help)
- >3 characters are often >1 word

Sequence length	# of characters	% of corpus
1 - 3 kanji	20,405,486	25.6
4 - 6 kanji	12,743,177	16.1
more than 6 kanji	3,966,408	5.1
Total	37,115,071	46.8

Figure 1: Statistics from 1993 Japanese newswire (NIKKEI), 79,326,406 characters total.

What's Coming in this Paper?

- Use of statistical analysis only, no language
- No rules specific to Japanese
- Requires very few (>=5) labeled training examples
- Requires large amounts of unsegmented data
- For long kanji strings, performance rivals current morphological models

How it Works

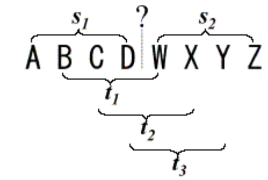


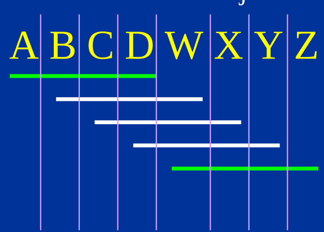
Figure 2: Collecting evidence for a word boundary – are the non-straddling n-grams s_1 and s_2 more frequent than the straddling n-grams t_1, t_2, and t_3?

Is $[\#(s_i) > \#(t_j)]$?

Calculates n-gram frequency over training corpus

How it Works (N=4)

Is $[\#(s_i) > \#(t_j)]$?



There are 5 4-grams in this sequence. With grouping, there are 2 X 3 = 6 greater-than expressions to evaluate

How it Works

Select which integers n ∈ N, for calculations of n-grams, do math, then determine word boundaries.

$$v_n(k) = \frac{1}{2(n-1)} \sum_{i=1}^{2}\sum_{j=1}^{n-1} I_>(\#(s_i^n), \#(t_j^n))$$

Then, we average the contributions of each n-gram order:

$$v_N(k) = \frac{1}{|N|} \sum_{n \in N} v_n(k)$$

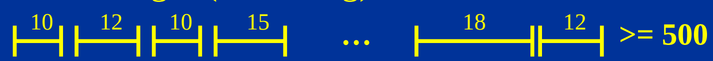
After $v_N(k)$ is computed for every location, boundaries are placed at all locations ℓ such that either:

- $v_N(\ell) > v_N(\ell - 1)$ and $v_N(\ell) > v_N(\ell + 1)$ (that is, ℓ is a local maximum), or
- $v_N(\ell) \geq t$, a threshold parameter.

Figure 3: Determining word boundaries. The X- Y boundary is created by the threshold criterion, the other three by the local maximum condition.

Experimental Methods

- Data from 150 MB Nikkei newswire 1993
- Pick 5 Held-out sets. Each…
  - 50 random chosen kanji sequences of length >=10 in length (12 on avg)



- Annotate held-out sets. Divide each into a parameter-training (50) and test (450) set

Segmenting Rules

- Word level
  - 1 word: (prefix+word+suffix)
- Morpheme level
  - 3 words: (prefix)(word)(suffix)

[小学校] [屋内] [[運動] [場]] [建設]

- 3 people had 98.42% agreement, all disagreement at morpheme level

Methods

- Morphological algorithms to compare to:
  - have access to lexicons of size 115,000 and 231,000.
  - used training data by adding it to their lexicons
- Parameters for the current method

  N = power set {2-6}

  l = .05k | 0 <= k <= 20

Evaluation

- Precision: "percentage of proposed brackets that exactly match word-level brackets in the annotation"

  = (# brackets right)/(#brackets proposed)
- Recall: "percentage of word-level annotation brackets that are proposed by the algorithm
  - = (# brackets right)/(#actual brackets)
- F-measure = 2PR / (P + R)

Segmentation Results

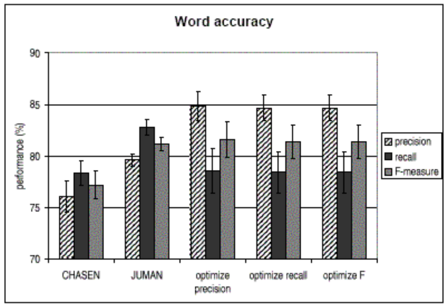


Figure 4: Word accuracy. The three rightmost groups represent our algorithm with parameters tuned for different optimization criteria.

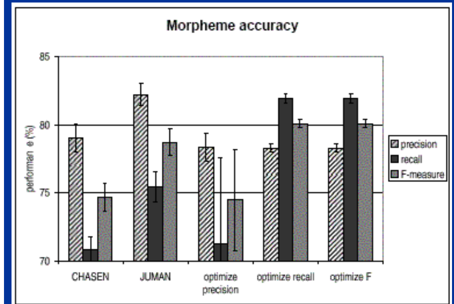


Figure 5: Morpheme accuracy.

Incompatible? Use New Metrics

- Crossing Bracket – “a proposed bracket that overlaps but is not contained within an annotation bracket”
- Morpheme Dividing Bracket – “subdivides a morpheme level annotation bracket”
- Compatible Brackets – neither of the above
- All-Compatible Brackets – sequence ratio of all correct

`[[data][base]][system]` **(annotation brackets)**

Proposed segmentation	word errors	morpheme errors	compatible-bracket errors: crossing	compatible-bracket errors: morpheme-dividing
`[data][base] [system]`	2	0	0	0
`[data][basesystem]`	2	1	1	0
`[database] [sys][tem]`	2	3	0	2

Figure 6: Examples of word, morpheme, and compatible-bracket errors. The sequence “data base” has been annotated as “[[data][base]]” because “data base” and “database” are interchangeable.

Results with new Metrics

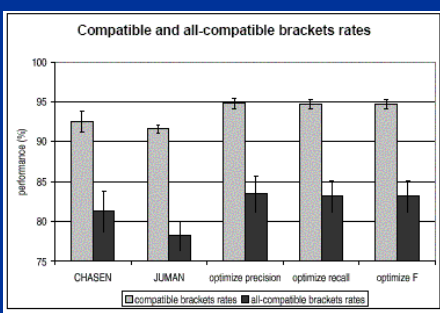


Figure 7: Compatible brackets and all-compatible bracket rates when word accuracy is optimized.

Discussion – Manual Effort

- Required Annotation
  - only the 50-sequence held-out sets (42min)
  - other methods require 1000-190,000 sentences
- Authors had some success with as few as only 5 sequences (4min)

	Juman5 vs. Juman50	Our50 vs Juman50	Our5 vs. Juman5	Our5 vs. Juman50
precision	-1.04	+5.27	+6.18	+5.14
recall	-0.63	-4.39	-3.73	-4.36
F-measure	-0.84	+0.26	+1.14	+0.30

Figure 8: Relative word accuracy as a function of training set size. “5” and “50” denote training set size *before* discarding overlaps with the test sets.

My Thoughts

- Purely Statistical Models are New
- This could work for other languages (Chinese), but would it do English well?
- The ‘>’ heuristic: “conjecture that using absolute differences may have an adverse effect”

Summary

- Purely Statistical Model
  - No lexicon or grammar
- Good Performance
  - Almost as good as, if not better than, other systems
- New Metrics