Recognizing Noisy Romanized Japanese Words in Learner English

Ryo Nagata†, Jun-ichi Kakegawa‡,
Hiromi Sugimoto*, Yukiko Yabuta*

†Konan University,
‡Hyogo University of Teacher Education,
*Japan Institute for Educational Measurement, Inc.

shshi

Romanized Japanese words

Roman words

mathya

Ainori
Asagaya
ganbaro
Monjya
Gzyunotou

shshi (*sushi*)

Romanized Japanese words

Roman words

mathya (*green tea*)

Ainori (name of TV program)
Asagaya (name of place)
ganbaro (*work hard*)
Monjya (kind of food)
Gzyunotou (five story pagoda)

Task: Recognizing Roman Words

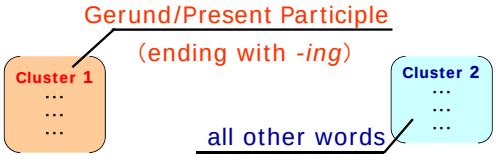
- Target: text written by Learners of English
  - contain many Roman words (20% of diff. words)
    → decreases performance of NLP systems
- Major Obstacle to overcome
  - Learner English contains spelling errors
  - Spelling rules are often violated
    e.g., because → becaus, becose, becoue, becauese, becuse, becaes, becase, becaues

Initial (but failed) Idea

- By clustering algorithm
  - *K*-means clustering
  - Both words have different spelling systems
  - Feature: trigram based
    attribute: trigram
    value: occurrence of trigram

Results are. . .

- worse than random guess
- Example of resulting clusters

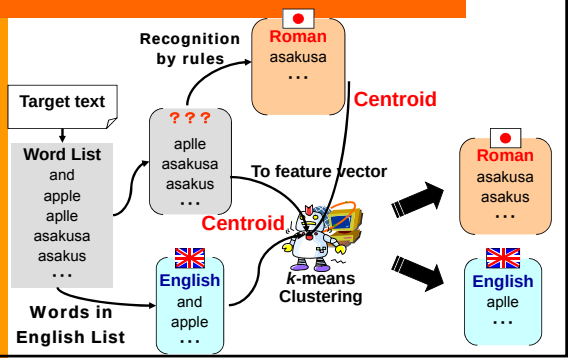


difficult to cover all English words by a cluster

Reconsideration of the Idea

- Observation
  - Roman words have different spelling system
    rule 1: Roman words end with a vowel or *n*
    rule 2: A consonant is followed by a vowel
- The problem is spelling errors
  - the two rules would perfectly recognize Roman words if there were no spelling errors
    rules + clustering

Proposed Method



Recognition by Rules

- Word to CV (Consonant Vowel) Pattern
  - e.g., SAMURAI → CVCVCVCV
    fighter → CVCCCVC
- Recognition using pattern matching
  ^[Vn]*(C[Vn]+)*$
  ≒sequences of CV, ends with V or n → Roman word

Word to Feature Vector

- Word to trigrams
  - e.g., SUSHI
    ^^S ^SU SUS USH SHI HI$ I$$
    ^: beginning of word
    $: end of word
- Trigram to vector
  - attribute: trigram
  - value: occurrence of trigram

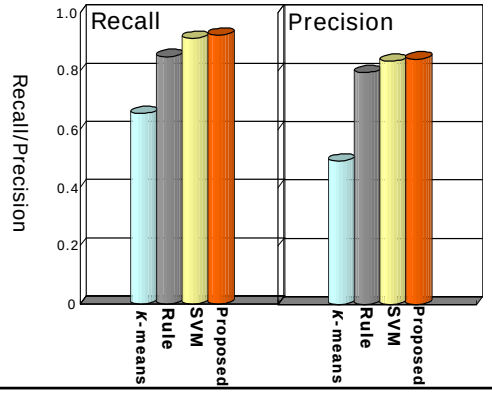
Evaluation

- Target essays
  - Writer：Jr. high
  - 117270 words
  - Number of different Roman words: 727
- Compared to
  - *K*-means clustering, Rule-based, SVMs
- English word list (20,000 words)
  - BNC（+10回/M words）& Ispell dictionary
- Performance measure: Recall, Precision

Training Data for SVM

- Roman instaces
  - From a Japanese dictionary
  - Pronunciation entry to Roman words (using a transliteration tool KAKASHI)
  - Number of instances: 160000
- English instances
  - From the English word list
  - Number of instances: 20000

Experimental Results



Disscussion

- Simple *k*-means clustering
  - does not work well
- Rule-based method performs well
  - Room for improvement
- SVMs outperform rule-based method
  - Training data do not cover misspelled words
- Proposed method performs equally/better
  - Initial centroids are obtained by rules
    → it adaptively learns plausible clusters

Characteristic Trigrams

Roman	English
i$$	y$$
u$$	s$$
ji$	d$$
aku	t$$
hi$	ed$
uji	r$$
^ko	g$$
^ka	l$$
ku$	ng$
ki$	^co

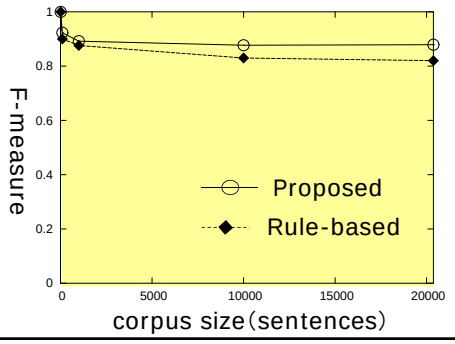
sorted by $\log(\frac{r_i}{e_i})$

$\frac{r_i}{e_i}$ value of *i*th trigram

(*e* denotes English centroid)
(*r* denotes Roman word centrod)

^:beginning of word
$:end of word

Relation between size and performance



Analyzing False Negatives and Positives

- False negatives
  - words consisting of English syllable or word
    e.g., omiyage (souvenir) → om, age
    English word: omnipotent, age
- False positive
  - misspelled words(94% of false positives)
  - Foreign words that follow spelling rules of Roman words
    e.g., pizza

Conclusions

- A method for recognizing Roman words
  - step 1: obtain initial centroids by some rules
  - step 2: *k*-means clustering
- Advantages of proposed method
  - robust against spelling errors
  - requires only an English word list
- A tool based on the proposed method:
  http://www.ai.info.mie-u.ac.jp/~nagata/tools/