Cross-Language IR at University of Tsukuba

Automatic Transliteration for Japanese, English, and Korean

Atsushi Fujii and Tetsuya Ishikawa
University of Tsukuba

C26

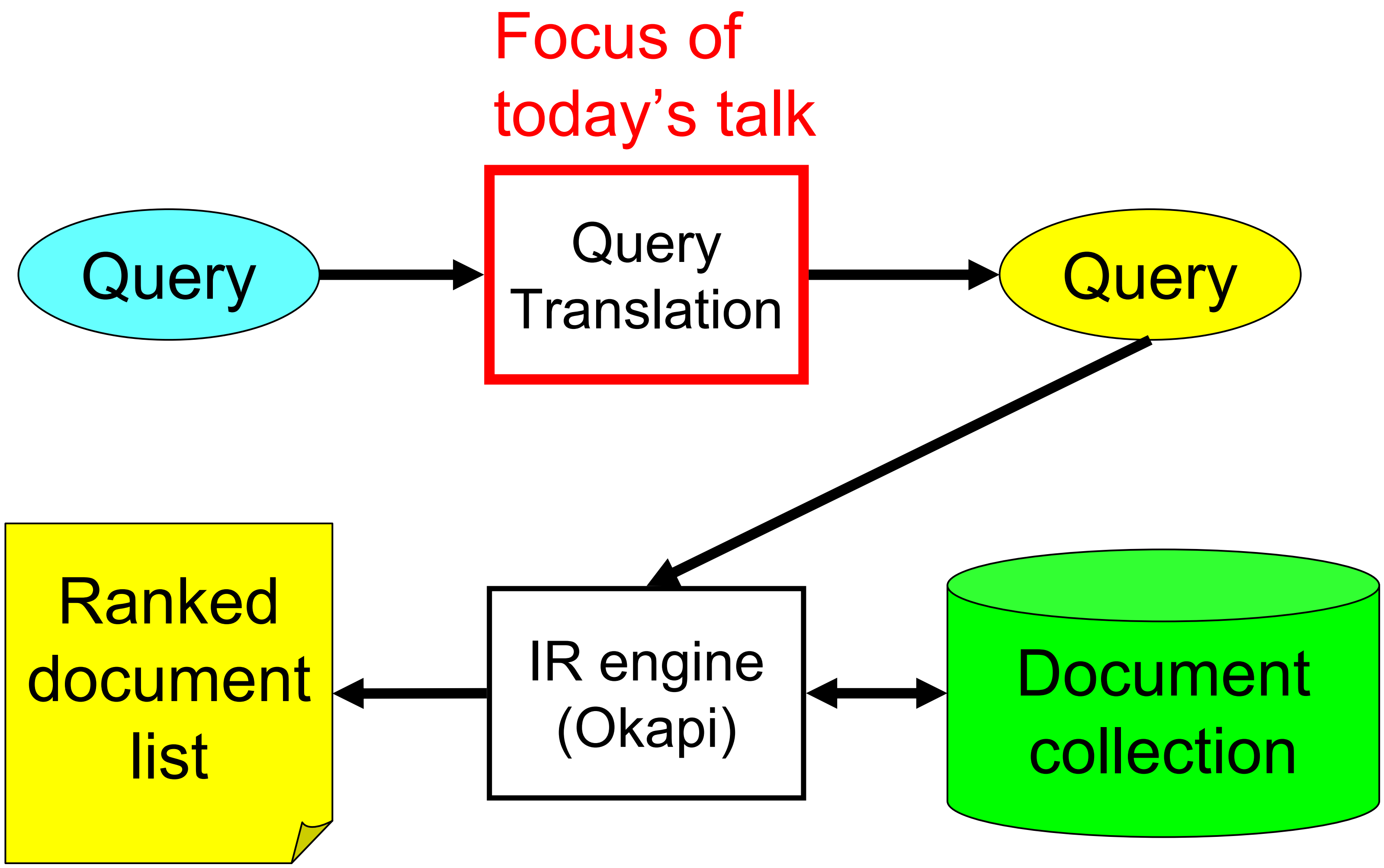
Motivation

- We developed an automatic transliteration method for Japanese and English CLIR
  - effective in translating foreign words spelled out by phonetic alphabet (e.g., Katakana)
  - evaluation since NTCIR-1
  - the method has been used in commercial cross-language patent IR service
- In NTCIR-4 CLIR, we applied our method to Korean and realized JEK transliteration in a single framework

Basis of transliteration

- spelling out foreign words (loanwords) by phonetic alphabet
  - technical terms and proper names
  - often out-of-dictionary words
- examples
  - dioxin → ダイオキシン，다이옥신
  - Yugoslavia → ユーゴスラビア，유고슬라비아
- back-transliteration
  - process to identify the source English word

Overview of our CLIR system



Example of J-E Query Translation

レジスタ転送言語

consulting dictionary
lexical segmentation

レジスタ　転送　言語

transliteration

resister
resistor
register

transfer
transmission
transport

language

disambiguation

register transfer language

Query Translation (cont.)

- compound query term S and a translation candidate T

  S = s1, s2, …, sN

  T = t1, t2, …, tM

  si and ti are base words

- compute P(T|S) = P(S|T)·P(T)

  P(S|T): translation model — P(T): language model

- select the candidate with max P(T|S)

Translation model

- $P(S|T) = \prod P(si \mid ti)$

  si and ti are base words comprising S and T

- heuristics and EM algorithm to correspond dictionary entries on a word-by-word basis

情報検索システム	Information retrieval system
検索モデル	retrieval model
情報抽出システム	Information extraction system
特許情報処理	patent information processing

- estimate $P(si \mid ti)$

Language model

- word-based trigram model
- 100K vocabulary in a target document collection
- Palmkit was used
  - compatible with CMU-LM toolkit

Transliteration method

- out-of-dictionary word S and a transliteration candidate T

  S = s1, s2, …, sN

  T = t1, t2, …, tM

  si and ti are letters (substrings of words)

- compute $P(T|S) = \underline{P(S|T)} \cdot \underline{P(T)}$

  transliteration model ← $P(S|T)$

  language model (word unigram) ← $P(T)$

- select the candidate with max $P(T|S)$

Transliteration dictionary

- dictionary for transliteration includes correspondence b/w source and target words on a phonogram-by-phonogram basis
- we use Roman representation as a pivot

Producing J/E dictionary

1. extract Japanese Katakana words and English translations from J-E dictionary
2. romanize Katakana words
3. correspond romanized Katakana and English words on a letter-by-letter basis
4. find the best correspondence

Example matrix

テキスト（te-ki-su-to）　text

	テ	キ	ス	ト	$
t	3	1	2	3	0
e	0	0	0	0	0
x	1	2	1	1	0
t	3	1	2	3	0
$	0	0	0	0	3

テ　te
キス　x
ト　t

By performing the same process for all Katakana entries, we produce transliteration dictionary

Extension to other languages

- our transliteration method can be applied to any language if represented by Roman characters

- no existing method has been used and evaluated in CLIR for more than two languages
  - our experiment was the first effort to explore this issue

Problems in Korean

- romanization of Korean words is more difficult than that of Katakana words
  - # of Hangul characters is approx. 11,000
  - one-to-one mapping b/w Hangul and Roman characters is not easy
- both conventional Korean words and foreign words are written by Hangul characters
  - detection of foreign words in Korean dictionary is crucial

Romanizing Korean words

- Hangul character consists of three types of consonants
  - – first consonant (19)
  - – vowel (21)
  - – last consonant (27 + 1)

last consonant is optional

- # of possible combinations is 11,172

(# of common characters is approx. 2,000)

- We used Unicode, in which characters are coded according to consonants

Fragment of Unicode table

- first consonant changes every 21 lines
- vowel changes every line and repeats every 21 lines
- last consonant changes every column

		ㄱ ㄲ ㄳ ㄴ ㄵ ㄶ ㄷ ㄹ ㄺ ㄻ ㄼ ㄽ ㄾ ㄿ ㅀ ㅁ ㅂ ㅄ ㅅ ㅆ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ	
가	44032:	가각갂갃간갅갆갇갈갉갊갋갌갍갎갏감갑값갓갔강갖갗갘같갚갛	:44059
개	44060:	개객갞갟갠갡갢갣갤갥갦갧갨갩갪갫갬갭갮갯갰갱갲갳갴갵갶갷	:44087
까	44620:	까깍깎깏깐깑깒깓깔깕깖깗깘깙깚깛깜깝깞깟깠깡깢깣깤깥깦깧	:44647
깨	... 44648:	깨깩깪깫깬깭깮깯깰깱깲깳깴깵깶깷깸깹깺깻깼깽깾깿꺀꺁꺂꺃	... :44675
나	45208:	나낙낚낛난낝낞낟날낡낢낣낤낥낦낧남납낪낫났낭낮낯낰낱낲낳	:45235
내	... 45236:	내낵낶낷낸낹낺낻낼낽낾낿냀냁냂냃냄냅냆냇냈냉냊냋냌냍냎냏	... :45263
	...		...

- Hangul characters can be identified by pronunciation
- only map b/w consonants and Roman characters is needed

Fragment of Unicode table

- first consonant changes every 21 lines
- vowel changes every line and repeats every 21 lines
- last consonant changes every column

		ㄱ ㄲ ㄳ ㄴ ㄵ ㄶ ㄷ ㄹ ㄺ ㄻ ㄼ ㄽ ㄾ ㄿ ㅀ ㅁ ㅂ ㅄ ㅅ ㅆ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ	
가	44032:	가각갂갃간갅갆갇갈갉갊갋갌갍갎갏감갑값갓갔강갖갗갘같갚갛	:44059
개	44060:	개객갞갟갠갡갢갣갤갥갦갧갨갩갪갫갬갭갮갯갰갱갲갳갴갵갶갷	:44087
까	44620:	까깍깎깏깐깑깒깓깔깕깖깗깘깙깚깛깜깝깞깟깠깡깢깣깤깥깦깧	:44647
깨	... 44648:	깨깩깪깫깬깭깮깯깰깱깲깳깴깵깶깷깸깹깺깻깼깽깾깿꺀꺁꺂꺃	... :44675
나	45208:	나낙낚낛난낝낞낟날낡낢낣낤낥낦낧남납낪낫났낭낮낯낰낱낲낳	:45235
내	... 45236:	내낵낶낷낸낹낺낻낼낽낾낿냀냁냂냃냄냅냆냇냈냉냊냋냌냍냎냏	... :45263
	...		...

- Hangul characters can be identified by pronunciation
- only map b/w consonants and Roman characters is needed

Fragment of Unicode table

- first consonant changes every 21 lines
- vowel changes every line and repeats every 21 lines
- last consonant changes every column

			ㄱ	ㄲ	ㄳ	ㄴ	ㄵ	ㄶ	ㄷ	ㄹ	ㄺ	ㄻ	ㄼ	ㄽ	ㄾ	ㄿ	ㅀ	ㅁ	ㅂ	ㅄ	ㅅ	ㅆ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	
가	44032:	가	각	갂	갃	간	갅	갆	갇	갈	갉	갊	갋	갌	갍	갎	갏	감	갑	값	갓	갔	강	갖	갗	갘	같	갚	갛	:44059
개	44060:	개	객	갞	갟	갠	갡	갢	갣	갤	갥	갦	갧	갨	갩	갪	갫	갬	갭	갮	갯	갰	갱	갲	갳	갴	갵	갶	갷	:44087
	...																													...
까	44620:	까	깍	깎	깏	깐	깑	깒	깓	깔	깕	깖	깗	깘	깙	깚	깛	깜	깝	깞	깟	깠	깡	깢	깣	깤	깥	깦	깧	:44647
깨	44648:	깨	깩	깪	깫	깬	깭	깮	깯	깰	깱	깲	깳	깴	깵	깶	깷	깸	깹	깺	깻	깼	깽	깾	깿	꺀	꺁	꺂	꺃	:44675
	...																													...
나	45208:	나	낙	낚	낛	난	낝	낞	낟	날	낡	낢	낣	낤	낥	낦	낧	남	납	낪	낫	났	낭	낮	낯	낰	낱	낲	낳	:45235
내	45236:	내	낵	낶	낷	낸	낹	낺	낻	낼	낽	낾	낿	냀	냁	냂	냃	냄	냅	냆	냇	냈	냉	냊	냋	냌	냍	냎	냏	:45263
	...																													...

- Hangul characters can be identified by pronunciation
- only map b/w consonants and Roman characters is needed

Fragment of Unicode table

- first consonant changes every 21 lines
- vowel changes every line and repeats every 21 lines
- last consonant changes every column

			ㄱ	ㄲ	ㄳ	ㄴ	ㄵ	ㄶ	ㄷ	ㄹ	ㄺ	ㄻ	ㄼ	ㄽ	ㄾ	ㄿ	ㅀ	ㅁ	ㅂ	ㅄ	ㅅ	ㅆ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	
가	44032:	가	각	갂	갃	간	갅	갆	갇	갈	갉	갊	갋	갌	갍	갎	갏	감	갑	값	갓	갔	강	갖	갗	갘	같	갚	갛	:44059
개	44060:	개	객	갞	갟	갠	갡	갢	갣	갤	갥	갦	갧	갨	갩	갪	갫	갬	갭	갮	갯	갰	갱	갲	갳	갴	갵	갶	갷	:44087
까	44620:	까	깍	깎	깏	깐	깑	깒	깓	깔	깕	깖	깗	깘	깙	깚	깛	깜	깝	깞	깟	깠	깡	깢	깣	깤	깥	깦	깧	:44647
깨	... 44648:	깨	깩	깪	깫	깬	깭	깮	깯	깰	깱	깲	깳	깴	깵	깶	깷	깸	깹	깺	깻	깼	깽	깾	깿	꺀	꺁	꺂	꺃	... :44675
나	45208:	나	낙	낚	낛	난	낝	낞	낟	날	낡	낢	낣	낤	낥	낦	낧	남	납	낪	낫	났	낭	낮	낯	낰	낱	낲	낳	:45235
내	... 45236:	내	낵	낶	낷	낸	낹	낺	낻	낼	낽	낾	낿	냀	냁	냂	냃	냄	냅	냆	냇	냈	냉	냊	냋	냌	냍	냎	냏	... :45263
	...																													...

- Hangul characters can be identified by pronunciation
- only map b/w consonants and Roman characters is needed

Detecting foreign words in Korean

- compute the phonetic similarity b/w romanized Hangul words and their translations (either English or Japanese)
- discard translation pairs whose similarity is below a threshold
  - conventional Korean words are discarded
- foreign word entries remained

Experiments (J/E)

<TITLE>, mean average precision (rigid)

Languages	#Entries	w/o transliteration	w/ transliteration
J-E	1M	0.2174	< 0.2182
E-J	1M	0.1250	= 0.1250
J-E (EDICT)	108K	0.1147	< 0.1383
E-J (EDICT)	108K	0.0612	< 0.0857

transliteration was effective for small dictionaries

Experiments (Korean)

<TITLE>, mean average precision (rigid)

Languages	w/o transliteration		w/ transliteration
J-K	0.2177	<	0.2457
K-J	0.1486	<	0.1746
E-K	0.2026	<	0.2153
K-E	0.1017	<	0.1231

transliteration was also effective for Korean

Conclusion

- realized transliteration for Japanese, English, and Korean in a single framework
- evaluated its effectiveness in NTCIR-4 CLIR task