Language Detection Library

- 99% over precision for 49 languages -

12/3/2010

Nakatani Shuyo @ Cybozu Labs, Inc.

What languages are these?

sprogregistrering

språkgjenkjenning

What languages are these?

sprogregistrering

Danish

språkgjenkjenning

Norwegian

What languages are these?

الكشف عن اللغة

تشخیص زبان

زبان کی شناخت

What languages are these?

الكشف عن اللغة — Arabic

تشخیص زبان — Persian

زبان کی شناخت — Urdu

What languages are these?

Language Detection

言語判別

What languages are these?

Language Detection

English

言語判別

Japanese

What's "Language Detection"?

- Detect language in which texts are written
  - also character code detection (excluded)
  - alias: Language identification / Language guessing

Japanese	English	Chinese
German	Spanish	Italian
Arabic	Hindi	Korean

Why Language Detection?

- Purpose
  - For language of search criteria
    - Query “Java” => Hit Chinese texts...
  - For SPAM filter/Extract content filter
    - To use language-specific information(punctuations, keywords)
- Usage
  - Web search engine
    - Apache Nutch bundles a language detection module
  - Bulletin board
    - Post in English, Japanese and Chinese

Methods

- The more languages, the more difficult
  - Among languages with the same script
  - Requires knowledge of scripts and languages
- A simple method:
  - Matching with the dictionary in each language
    - Huge dictionary(inflections, compound words)
- Our method:
  - Calcurates language probabilities from features of spelling
    - Naive Bayse with character n-gram

Existing Language Detection

- There are a few libraries of language detection.
  - Usage was limited?
    - For only web search?
    - But all services will become global from now on!
  - Building corpus/model is a expensive work.
    - Requires knowledge of scripts and languages
- Few languages supported & low precision
  - Almost 10 languages. Not including Asian ones

“Practical” Language Detection

- 99% over precision
  - 90% is not practical. (100 of 1000 mistakes)
- 50 languages supported
  - European, Asian and so on
- Fast Detection
  - Many documents available
- Output each language’s probability
  - For multiple candidates

Language Detection Library for Java

- We developed a language detection library for Java.
  - Generates the language profiles from training corpus
    - Profile : the probabilities of all spellings in each language
  - Returns the candidates and their probabilities for given texts
  - 49 languages supported
- Open Source (Apache License 2.0)
  - http://code.google.com/p/language-detection/

Experiments

- Training
  - 49 languages from Wikipedia
    - That can provide a test corpus of its language
- Test
  - 200 news articles of 49 languages
    - Google News (24 languages)
    - News sites in each language
    - Crawling by RSS

Results (1)

languages		#	precisions	items
af	Afrikaans	200	199 (99.50%)	en=1, af=199
ar	Arabic	200	200 (100.00%)	ar=200
bg	Bulgarian	200	200 (100.00%)	bg=200
bn	Bengali	200	200 (100.00%)	bn=200
cs	Czech	200	200 (100.00%)	cs=200
da	Danish	200	179 (89.50%)	da=179, no=14, en=7
de	German	200	200 (100.00%)	de=200
el	Greek	200	200 (100.00%)	el=200
en	English	200	200 (100.00%)	en=200
es	Spanish	200	200 (100.00%)	es=200
fa	Persian	200	200 (100.00%)	fa=200
fi	Finnish	200	200 (100.00%)	fi=200
fr	French	200	200 (100.00%)	fr=200
gu	Gujarati	200	200 (100.00%)	gu=200
he	Hebrew	200	200 (100.00%)	he=200
hi	Hindi	200	200 (100.00%)	hi=200
hr	Croatian	200	200 (100.00%)	hr=200
hu	Hungarian	200	200 (100.00%)	hu=200
id	Indonesian	200	200 (100.00%)	id=200
it	Italian	200	200 (100.00%)	it=200
ja	Japanese	200	200 (100.00%)	ja=200
kn	Kannada	200	200 (100.00%)	kn=200
ko	Korean	200	200 (100.00%)	ko=200
mk	Macedonian	200	200 (100.00%)	mk=200
ml	Malayalam	200	200 (100.00%)	ml=200

Results (2)

languages		#	precisions		items
mr	Marathi	200	200	(100.00%)	mr=200
ne	Nepali	200	200	(100.00%)	ne=200
nl	Dutch	200	200	(100.00%)	nl=200
no	Norwegian	200	199	(99.50%)	da=1, no=199
pa	Punjabi	200	200	(100.00%)	pa=200
pl	Polish	200	200	(100.00%)	pl=200
pt	Portuguese	200	200	(100.00%)	pt=200
ro	Romanian	200	200	(100.00%)	ro=200
ru	Russian	200	200	(100.00%)	ru=200
sk	Slovak	200	200	(100.00%)	sk=200
so	Somali	200	200	(100.00%)	so=200
sq	Albanian	200	200	(100.00%)	sq=200
sv	Swedish	200	200	(100.00%)	sv=200
sw	Swahili	200	200	(100.00%)	sw=200
ta	Tamil	200	200	(100.00%)	ta=200
te	Telugu	200	200	(100.00%)	te=200
th	Thai	200	200	(100.00%)	th=200
tl	Tagalog	200	200	(100.00%)	tl=200
tr	Turkish	200	200	(100.00%)	tr=200
uk	Ukrainian	200	200	(100.00%)	uk=200
ur	Urdu	200	200	(100.00%)	ur=200
vi	Vietnamese	200	200	(100.00%)	vi=200
zh-cn	Simplified Chinese	200	200	(100.00%)	zh-cn=200
zh-tw	Traditional Chinese	200	200	(100.00%)	zh-tw=200
sum		9800	9777	(99.77%)	

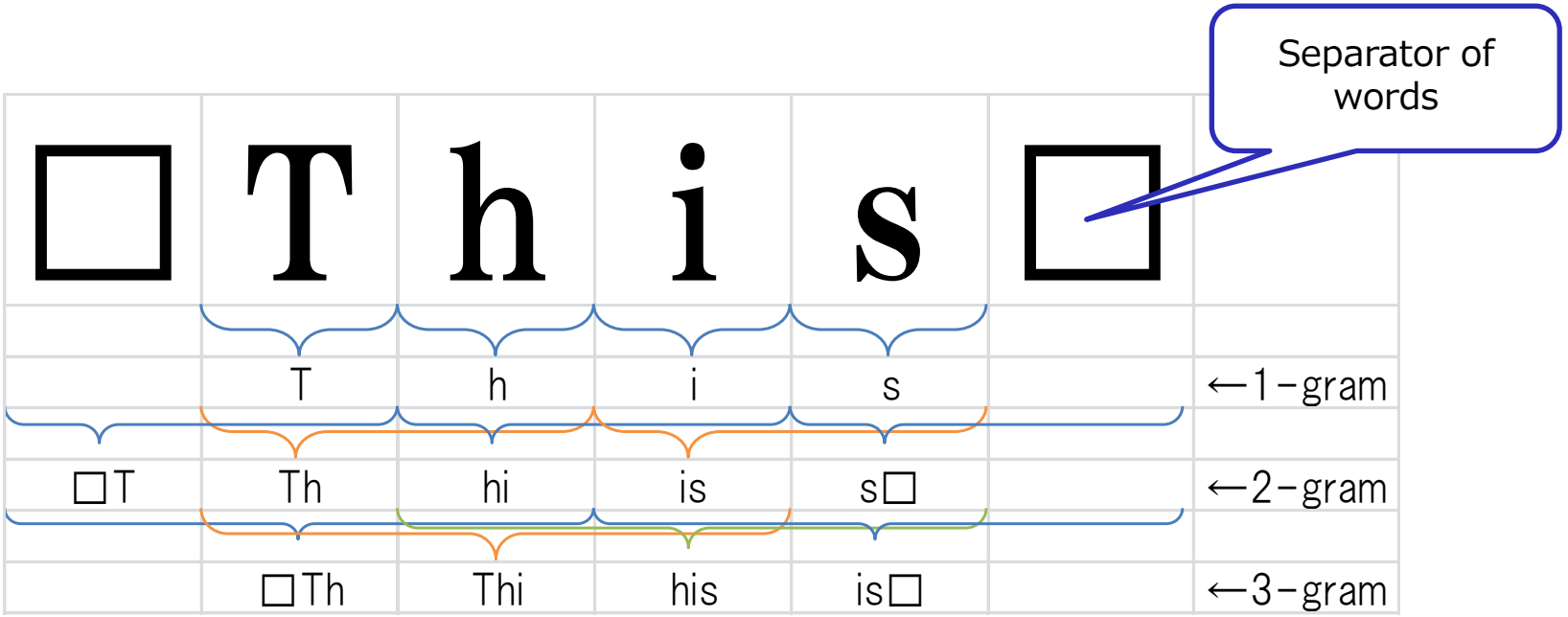
Algorithms

Language Detection with Naive Bayes

- Classifies documents into “language” categories
  - Categories: English, Japanese, Chinese, …
- Updates the posterior probabilities of categories by feature probabilities in each category
  - $p(C_{\mathrm{k}}|X)^{(\mathrm{m}+1)} \propto p(C_{\mathrm{k}}|X)^{(\mathrm{m})} \cdot p(X_i|C_k)$
    - where C_{k}:category, X:document, X_i:feature of document
  - Terminates detection process if the maximum probability(normalized) is over 0.99999
    - Early termination for perfomance

Features of Language Detection

- Character n-gram
  - To be exact, “Unicode’s codepoint n-gram”
  - Much less than the size of words



How to detect the text's language

- Each language has the peculiar characters and spelling rule.
  - The accented "é" is used in Spanish, Italian and so on, and not used in English in principle.
  - The word that starts with "Z" is often used in German and rarely used in English.
  - The word that starts with "C" and contains spell "Th" are used in English and not used in German.
- Accumulates the probabilities assigned to these features in given text, so the guessed language is obtained as one that has the maximum probability.

	□C	□L	□Z	Th
English	0.75	0.47	0.02	0.74
German	0.10	0.37	0.53	0.03
French	0.38	0.69	0.01	0.01

Improvement for Naive Detection

- The above naive algorithm can detect only 90% precision.
  - Not “practical”
  - Very low precision for some languages
    - Japanese, Traditional Chinese, Russian, Persian, ...
- Cause:
  - Bias and noise of training and test corpus
- Improvement
  - Noise filter
  - Character normalization

(1) Bias of Characters

- Alphabet / Arabic / Devanagari
  - About 30 characters
- Kanji (Chinese character)
  - 20000 characters over!
    - 1000 times as much as Alphabets
- Kanji has “zero frequency problem”
  - Can’t detect language of “谢谢”(Simplified Chinese)
    - This character isn’t used on Wikipedia.
  - Name Kanji (uneven frequency)

Normalization with “Joyo Kanji”

- Classifies “similar frequency Kanji” and normalizes each cluster into a representative Kanji.
  - (1) Clustering by K-means
  - (2) Classification by “Joyo Kanji”
    - Joyo Kanji (常用漢字: regularly used Kanji)
    - Simplified Chinese: “现代汉语常用字表”(3500 characters)
    - Traditional Chinese: the first standard of Big5 (5401 characters). It includes “常用国字標準字体表” (4808 characters)
    - Japanese: Joyo Kanji(2136 characters) + the first standard of JIS X 0208 (2965 characters) = 2998 characters
  - 130 clusters
    - Each language has about 50 classes.

(2) Noise of Corpus

- Removes the language-independent characters
  - Numeric figures, symbols, URLs and mail addresses
- Latin character noise in non-Latin text
  - Alphabets often occur in also non-Latin text.
    - Java, WWW, US and so on
  - Remove all Latin-characters if their rate is less than 20%.
- Latin character noise in Latin text
  - Acronyms, person's names and place names don't represent feature of languages.
    - UNESCO, "New York" in French text
    - Person's name has a various language feature (e.g. Mc- = Gaelic).
  - Removes all-capital words
  - Reduces the effect of local features by the feature-sampling

Normalization of Arabic Character

- All Persian texts were detected as Arabic!
  - Persian and Arabic belong to different language families, so it ought to be easy to discriminate them.
- A high-frequency character “yeh” is assigned to different codes in training and test corpora respectively.
  - In the training corpus (Wikipedia), it is assigned to “ى” (¥u06cc, Farsi yeh).
  - In the test corpus (News), it is “ي” (¥u064a, Arabic yeh).
    - Cause: Arabic character-code CP-1256 don’t has the character mapped to ¥u06cc, so it is substituted to ¥u064a in a general way.
- Normalizes ¥u06cc(Farsi yeh) into ¥u064a(Arabic yeh)
  - All Persian texts are detected correctly.

Conclusion

Conclusion

- We developed the language detection library for Java.
  - 49 languages can be detected in 99.8% precision.
  - Our next product will use it (search by language).
- 90% is easy. But 99% over is practical.
  - Ideal: Answer from the novel beautiful theory
  - Real: Unrefined steady way all along

Open Issues

- Short text (e.g. twitter)
- Arabic vowel signs
- Text in more than one language
- Source code in text

References

- [Habash 2009] Introduction to Arabic Natual Language Processing
  - http://www.medar.info/conference_all/2009/Tutorial_1.pdf
- [Dunning 1994] Statistical Identification of Language
- [Sibun & Reynar 1996] Language identification: Examining the issues
- [Martins+ 2005] Language Identification in Web Pages
- 千野栄一編「世界のことば100語辞典 ヨーロッパ編」
- 町田和彦編「図説 世界の文字とことば」
- 世界の文字研究会「世界の文字の図典」
- 町田和彦「ニューエクスプレス　ヒンディー語」
- 中村公則「らくらくペルシャ語　文法から会話」
- 道広勇司「アラビア系文字の基礎知識」
  - http://moji.gr.jp/script/arabic/article01.html
- 北研二,辻井潤一「確率的言語モデル」

Thank you for reading

- Language Detection Project Home
  - http://code.google.com/p/language-detection/
- blog (in Japanese)
  - http://d.hatena.ne.jp/n_shuyo/
- twitter
  - http://twitter.com/shuyo