ISMLA Multilingual Session 3: Recognizing Foreign Names in Chinese

Johannes Dellert

Tübingen University

November 15, 2017

Plan

1. Foreign Names in Chinese
2. A Simple Algorithm
3. Exercise 02: Recognizing Foreign Names in Chinese

Recognizing Foreign Names: Motivation

Consider the following result of our greedy tokenizer:

摩	mó	VT	rub
泽	zé	N	pond
尔	ěr	PRN	thou
河	hé	N	river
和	hé	N	harmony
美	měi	EV	pretty
因	yīn	NK	reason
河	hé	N	river

Solution: (next slide)

Recognizing Foreign Names: Motivation

This is what a good tokenizer should output:

摩泽尔	mózéěr	NE	[mozel]
河	hé	N	river
和	hé	CNJ	and
美因	měiyīn	NE	[mein]
河	hé	N	river

Solution: the rivers Moselle and Main!

Writing Foreign Names in Chinese

- **problem:** in principle, every symbol has a meaning!
- any symbols used for phonetic transcription will have other functions in the language, i.e. detecting phonetic reading must rely on context
- **idea**: delineate a subset of Chinese characters which can be used for their phonetic value instead of their literal meaning
- **advantage**: there are no non-standard pronunciation rules as e.g. for loans in many Latin-script languages (c.f. German *Fauxpas*)
- **advantage**: if you leave some choice, you can play with the meanings (used in company names, e.g. 西门子 west-gate-master "Siemens")
- **disadvantage**: phonotactics are restricted to possible syllables of Mandarin Chinese (Xīménzǐ)

Writing Foreign Names in Chinese

- Chinese **news agency standard** for transcribing English:

Transcription from English (IPA) into Chinese																									
	-	b	p	d	t	g	k	v	w	f	z / dz	ts	s / ð / θ	ʒ	ʃ	dʒ	tʃ	h	m	n	l	r*	j	g^w	k^w	h^w
-		布	普	德	特	格	克	夫 / 弗			兹	茨	斯 / 丝	日	什	奇		赫	姆	恩	尔		伊	古	库	胡
ɑː, æ, ʌ	阿	巴 / 芭	帕	达	塔	加	卡	瓦 / 娃		法 / 娃	扎	察	萨 / 莎	扎	沙 / 莎	贾	查	哈	马 / 玛	纳 / 娜	拉		亚 / 娅	瓜	夸	华
ɛ, eɪ	埃	贝	佩	德	特 / 泰	盖	凯	韦		费	泽	策	塞	热	谢	杰	切	赫 / 黑	梅	内	莱	雷 / 蕾	耶	圭	奎	惠
ɜ, ə	厄	伯	珀	德	特	格	克	弗	沃	弗	泽	策	瑟	热	舍	哲	彻	赫	默	纳 / 娜	勒		耶	果	阔	霍
iː, ɪ	伊	比	皮	迪	蒂	吉	基	维	威	菲	齐		西	日	希	吉	奇	希	米	尼 / 妮	利 / 莉	里 / 丽	伊	圭	奎	惠
ɒ, ɔː, oʊ	奥 / 欧	博	波	多	托	戈	科	沃		福	佐	措	索	若	肖	乔		霍	莫	诺	洛	罗 / 萝	约	果	阔	霍
uː, ʊ	乌	布	普	杜	图	古	库	武	伍	富	祖	楚	苏	茹	舒	朱	楚	胡	穆	努	卢	鲁	尤		库	
juː, jʊ	尤	比尤	皮尤	迪尤	蒂尤	久	丘	维尤	威尤	菲尤	久	丘	休		休	久	丘	休	缪	纽	柳	留				
aɪ	艾	拜	派	代 / 戴	泰	盖	凯	韦	怀	法	宰	蔡	赛		夏	贾	柴	海	迈	奈	莱	赖	耶	瓜伊	夸	怀
aʊ	奥	鲍	保	道	陶	高	考	沃		福	藻	曹	绍		绍	焦	乔	豪	毛	瑙	劳		尧		阔	
æn, ʌn, æŋ	安	班	潘	丹	坦	甘	坎	万		凡	赞	灿	桑		尚	詹	钱	汉	曼	南	兰		扬	关	宽	环
ɑn, aʊn, ʌŋ, ɔn, ɒn, ɒŋ	昂	邦	庞	当	唐	冈	康	旺		方	藏	仓	桑	让		章	昌	杭	芒		朗			光	匡	黄
ɛn, ɛŋ, ɜn, ən, əŋ	恩	本	彭	登	滕	根	肯	文		芬	曾	岑	森	任	申	真	琴	亨	门	嫩	伦		延	古恩	昆	
ɪn, in, ɪən, jən	因	宾	平	丁	廷	金		温		芬	津	欣	辛		欣	金	钦	欣	明	宁	林 / 琳		因	古因	昆	
ɪŋ	英					京	金				京	青			兴	京	青	兴					英	古英		
un, ʊn, oʊn	温	本	蓬	敦	通	贡	昆	文		丰	尊	聪	孙		顺	准	春	洪	蒙	农	伦		云			
ʊŋ	翁 / 宏	邦		东			孔	翁			宗		松	容	雄	琼					隆	龙	永			洪

Chinese Transcription: Examples

Here are some country names, sometimes older than the standard:

- 阿富汗 Āfùhàn (“mountain-abundant-sweat”)
- 澳大利亚 Àodàlìyà (“inlet-big-profit-secondary”)
- 丹麦 Dānmài (“cinnabar-wheat”)
- 芬兰 Fēnlán (“fragrant-orchid”)
- 加拿大 Jiānádà (“add-take-big”)
- 新加坡 Xīnjiāpō (“new-add-slope”)
- 伊拉克 Yīlākè (“he-pull-overcome”)
- 意大利 Yìdàlì (“will-big-profit”)

Recognizing Foreign Names: An Algorithm

Some simple heuristics are relatively easy to implement:

- the longer a chunk of consecutive symbols with phonetic reading, the more likely it becomes that we are dealing with a name
- **idea**: treat chunks of two or more phonetic characters as names!
- **problem**: some frequent words are composed of such characters, e.g. 东南 dōngnán "southeast" and 地方 dìfang "place"
- **algorithm** (which you are going to implement):
  1. perform greedy tokenization without paying attention to phonetics
  2. add a second annotation layer of all chunks of possibly phonetic symbols
  3. for each chunk of possibly phonetic symbols:
     - check whether greedy tokenization has found meaningful multi-character words within the chunk
     - if yes, the chunk is unlikely to represent a name
     - if no, delete the monosyllabic tokens, assume you have found a name

Recognizing Foreign Names: Example

orth	包	括	雷	根	斯	堡
pinyin	bāo	kuò	léi	gēn	sī	bǎo
greedy	including		thunder	root	this	castle
phon	bau	?	re	gen	s	?
joint	including			[regens]		castle

“including Regensburg”

Exercise 02: The Expanded Dictionary

Lexical lookup in `CmnEngDictionaryWithPhon`:

- initialization: `new CmnEngDictionaryWithPhon()`
- use: `List<SimpleDictionaryEntry> lookup(String orthForm)`

The `SimpleDictionaryEntry` looks exactly as before, but:

- lookup can return a list of size > 1!
- there will be new one-character entries of category `PHON` for all symbols commonly used for foreign names; meaningful variants are represented separately (with non-`PHON` categories)
- the gloss list of each `PHON` entry will contain all the IPA sequences this character can represent according to the standard, plus a first entry which represents a best guess (concatenating them will give you things like [mozel] and [mein])

Exercise 02, Steps 1 and 2: Adapting the Greedy Tokenizer

Steps to perform on your greedy tokenizer:

- use `CmnEngDictionaryWithPhon` instead of `CmnEngDictionary`
- expand the type system by adding two subtypes of `TokenCMN`:
  - `MeaningfulTokenCMN` for meaningful tokens
  - `PhonChunkCMN` for chunks of phonetic symbols
- your greedy tokenizer needs to ignore `PHON` entries
- for this, it will need to search the returned list of `SimpleDictionaryEntry` objects (`get(0)` will not work any longer)
- it needs to produce `MeaningfulTokenCMN` annotations

Exercise 02, Step 3: `TranslitAnnotatorCMN`

A new primitive analysis engine should exhibit the following behavior:

- annotate as `PhonChunkCMN` instances consecutive chunks (length ≥ 2) of Chinese syllables which have a dictionary entry with `cat` value `PHON`
- `cat` value should be `PHON`
- gloss value should be concatenated from the first glosses provided for each syllable (here `get(0)` will work!) and put into square brackets
- the Pinyin value for the chunk should be just the concatenation of the syllables' Pinyin values
- NB: length ≥ 2 means that isolated `PHON` symbols are ignored!

Exercise 02, Step 4: `TokenAnnotationCombinerCMN`

Another new primitive analysis engine should unify the annotations:

- iterate through all `PhonChunkCMN` annotations
- for each `PhonChunkCMN` object, use the method `AnnotationIndex<MeaningfulTokenCMN>.subiterator(chunk)` to check whether it includes a `MeaningfulTokenCMN` of length ≥ 2
- if it does, remove the `PhonChunkCMN` annotation and leave the meaningful subtokens intact
- otherwise, remove all the meaningful subtokens, so that only the chunk remains annotated

Exercise 02, Step 5: `GlossOutputCMN`

Finally, we need a CAS Consumer which stores our analyses in files:

- the `TokenCMN` annotations of each document should be printed in default iteration order (i.e. by begin index) into a separate TSV file
- the output format should include the following columns: orthography, pinyin, category, and gloss
- the files should be generated in a directory that can be configured by a parameter

Exercise 02, Step 6: The aggregate analysis engine

The analysis engines are now chained together into a pipeline:

- define an aggregate analysis engine which calls the previously developed components in the following order:
  1. `GreedyTokenizerCMN`
  2. `TranslitAnnotatorCMN`
  3. `TokenAnnotationCombinerCMN`
  4. `GlossOutputCMN`

Run the aggregate engine on the Wikipedia snippets.

Exercise 02, Step 7: Analysing the Results

Programming is done, now let us work with the output a bit:

- Inspect the gloss output by reading the gloss column, see how well you can understand the content of the snippets. If you cannot interpret the syntax of some sentences, running them through Google Translate will help you (it works pretty well for Chinese).
- To see whether the named entities were correctly recognized, look some of them up in the Chinese Wikipedia (`zh.wikipedia.org`) until you hit some entities you cannot find in this way.
- There is a common pattern to these examples which has something to do with multi-segment tokens that overlap the boundaries of the `PhonChunkCMN`s. What is the problem? How could the algorithm be refined to handle this case?

Exercise 02: Hints

Things to pay attention to during implementation:

- you might need to process iterators and store their contents in an auxiliary data structure to avoid concurrent modification
- think carefully about the input and output capabilities of each analysis engine; incorrect specification can prevent annotation!

Hints for inspecting the data:

- Chinese word order differs from English in some respects, e.g. it uses postpositions instead of prepositions
- some words are ambiguous between noun and verb, but our greedy tokenizer cannot distinguish them