Regular Expressions in Python: Tagging Parts of Speech in Persian

LIN 511 – Computation Linguistics

Dr. Andrew Hippisley

Paul Eberhart

Ghazelah Kazeminejad

Drake Hyman

Lauren Ashburn

Kimberly Haney

Model

- Utilize Python/NLTK toolchain to create a Part of Speech Tagger to classify words from selected Persian passages
  - Three phases:
    - Define Mini-Lexicon
    - Regular Expression Tagger (NLTK)
    - Bigram Tagger (Hand Coded)
- Support Persian Script

Persian Language & Alphabet

- 32 characters, similar to Arabic alphabet
- Script is necessarily cursive due to ligatures
- Alphabet consists of multiple positional forms for each character
- Persian script is written/read from right-left

Name	DIN 3135	IPA	Contextual Forms			
			End	Middle	Beginning	Isolated
ʾalef	ā / ʾ	[ɒ], [ʔ]	ـا	ـا *	ا / آ *	ا
be	b	[b]	ـب	ـبـ	بـ	ب
pe	p	[p]	ـپ	ـپـ	پـ	پ
te	t	[t]	ـت	ـتـ	تـ	ت
s̱e	s̱	[s]	ـث	ـثـ	ثـ	ث

Persian Morphology

- SOV Word order
  - Objects are marked, and thus relocatable
- Half-Spaces, full-spaces (different meanings, can alter proximal characters)
- Compound Verbs, Adverbs, Prepositions
- No morphological marker to distinguish Nouns from Adjectives (must know lexicon)

Other features of Persian

- 3 kinds of plural markers, also found at ends of other words; may cause over generation
- Persian has no proper noun marker
- Short vowels are not drawn in text, so words like “of” disappear (similar to English understood pronouns)

Persian Language Challenges

- Tokenizing compounds is hard
  - No way for the tokenizer to distinguish whitespace in compounds from spaces between words
- A few nouns violate prefix/suffix rules
- Suffix “i” might be added to verbs,nouns, or adjectives

Unicode and UTF-8

- Standard for encoding different character sets
  - Supports > 110,000 characters/code points in >100 scripts
- Includes direction markers
  - Allows mixing of English (L-to-R) and Persian (R-to-L)
- Supports several different encodings
  - Most common is UTF-8 “Universal Character Set Transformation Format—8-bit”
    - One to Four 8-bit bytes per character
    - Overlaps with ASCII for Latin alphabet

Unicode Direction

- Unicode has special characters for direction changes
  - U+202A (&lrm), Left to Right Marker
  - U+202B (&rlm), Right to Left Marker
- BIDI (Bi-Directional Text) systems class letters as direction sensitive or neutral
  - Whitespace and symbols are neutral
- The first & last characters of a right to left regular expressions might be swapped .. depending on the editor

Python/NLTK

- Python has decent Unicode/UTF-8 support
  - Critically, Including the `re` library
- NLTK includes automated PoS tagger construction tools
- NLTK's UTF-8 Support is incomplete
  - Makes unicode-unsafe calls to re functions

New Regexp Tagger

One critical change on line 477

/usr/lib/python2.7/site-packages/nltk/tag/sequential.py

```
461     yaml_tag = '!nltk.RegexpTagger'
462
463     def __init__(self, regexps, backoff=None):
464         """
465         """
466         SequentialBackoffTagger.__init__(self, backoff)
467         labels = ['g'+str(i) for i in range(len(regexps))]
468         tags = [tag for regex, tag in regexps]
469         self._map = dict(zip(labels, tags))
470         regexps_labels = [(regex, label) for ((regex,tag),label) in zip(regexps,labels)]
471         self._regexs = re.compile('|'.join(['(?P<%s>%s)' % (label, regex) for regex,label in regexps_labels]))
472         self._size=len(regexps)
473         print "Using abused unicode version of sequential:RegexpTagger -PAPPP,20130423"
474
475     def choose_tag(self, tokens, index, history):
476         #print "Using abused version of sequential"
477         m = self._regexs.match(tokens[index].decode("utf-8"))
478         if m:
479           return self._map[m.lastgroup]
480         return None
```

Solution

- `RegExpTagger` had to be modified to work on Unicode strings, replacing the normal `RegexpTagger`
- Generate some Regular Expressions from Variables defining subwords
- Hand-code a simple Bigram pass

Demonstration!

شیوه عطسه کردن، اطلاعات زیادی را در مورد شخصیت انسان می‌دهد. این نکته از یافته های یک پژوهش تازه است. به گزارش سرویس پژوهشی خبرگزاری دانشجویان ایران، به گفته دکتر آلان هیرچ، متخصص مغز و اعصاب بنیاد پژوهش و درمان بویایی و چشایی، یک عطسه بلند و انفجاری احتمالاً از سوی یک شخصیت برونگرا و اجتماعی رخ می‌دهد، در حالیکه یک فرد خجالتی بیشتر تلاش می‌کند تا عطسه آرامتری داشته باشد.

Hand-Tagged Example

شیوه_NN عطسه_NN کردن_GND، اطلاعات_NNS زیادی_JJ را_OM در مورد_IN شخصیت_NN انسان_NN می‌دهد_VPres. این_DT نکته_NN از_IN یافته‌های_NNS یک_CD پژوهش_NN تازه_JJ است_VPres. به_IN گزارش_NN سرویس_NN پژوهشی_JJ خبرگزاری_NN دانشجویان_NNS ایران_NNP، به_IN گفته_NN دکتر_NN آلان_NNP هیرچ_NNP، متخصص_NN مغز_NN و_CC اعصاب_NN بنیاد_NN پژوهش_NN و_CC درمان_NN بویایی_NN و_CC چشایی_NN، یک_CD عطسه_NN بلند_JJ و_CC انفجاری_JJ احتمالاً_RB از سوی_IN یک_CD شخصیت_NN برونگرا_JJ و_CC اجتماعی_JJ رخ می‌دهد_VPres، در حالیکه یک_CD فرد_NN خجالتی_JJ بیشتر_JJR تلاش می‌کند_VPres تا_SC عطسه_NN آرامتری_JJR داشته باشد_V