NLTK and ibus-typing-booster

Presented by Mike Fabian

Ideas how to use NLTK to improve ibus-typing-booster

What is NLTK?

- “Natural Language Toolkit” a library for “Natural language processing” (NLP)
- Written in Python
- NLTK 3.0.0 (Python3 version) just released (2014-09-11) → usable by ibus-typing-booster

Interesting features for i-t-b

- Accessing corpora
- Parsing
- Tokenizing (better than my primitive code?)
- Stemming
- Collocation discovery
- Part-of-speech (POS) tagging
- Semantic interpretation

Synonyms

```
import nltk
from nltk.corpus import wordnet
>>> wordnet.synsets('car')
[Synset('car.n.01'), Synset('car.n.02'), Synset('car.n.03'), Synset('car.n.04'),
Synset('cable_car.n.01')]
>>> wordnet.synset('car.n.01').lemma_names()
['car', 'auto', 'automobile', 'machine', 'motorcar']
>>> wordnet.synset('car.n.02').lemma_names()
['car', 'railcar', 'railway_car', 'railroad_car']
>>> wordnet.synset('car.n.03').lemma_names()
['car', 'gondola']
>>> wordnet.synset('car.n.04').lemma_names()
['car', 'elevator_car']
>>> wordnet.synset('cable_car.n.01').lemma_names()
['cable_car', 'car']
```


Synonym example

```
def synonyms(word):
    u'''List synonyms for word

    >>> synonyms('fedora')
    ['Stetson', 'felt hat', 'homburg', 'trilby']
    '''
    return sorted(set(lemma_name.replace('_', ' ')
                      for synset in wordnet.synsets(word)
                      for lemma_name in synset.lemma_names()
                      if lemma_name.replace('_', ' ') != word))
```


Hyponym example

```
def hyponyms(word):
    u'''List hyponyms for word


    >>> hyponyms('hat')
    ['Panama', 'Panama hat', 'Stetson', 'bearskin', 'beaver', 'boater', 'bonnet',
'bowler', 'bowler hat', 'busby', 'campaign hat', 'cavalier hat', 'cocked hat', 'cowboy
hat', 'deerstalker', 'derby', 'derby hat', 'dress hat', 'dunce cap', "dunce's cap",
'fedora', 'felt hat', "fool's cap", 'fur hat', 'high hat', 'homburg', 'leghorn',
'millinery', 'opera hat', 'plug hat', 'poke bonnet', 'sailor', 'shako', 'shovel hat',
'silk hat', 'skimmer', 'slouch hat', 'snap-brim hat', 'sombrero', "sou'wester",
'stovepipe', 'straw hat', 'sun hat', 'sunhat', 'ten-gallon hat', 'tirolean', 'titfer',
'top hat', 'topper', 'toque', 'trilby', 'tyrolean', "woman's hat"]
    '''
    return sorted(set(lemma.name().replace('_', ' ')
                      for synset in wordnet.synsets(word)
                      for hyponym in synset.hyponyms()
                      for lemma in hyponym.lemmas()
                      if lemma.name().replace('_', ' ') != word)
```


Hypernym example

```
def hypernyms(word):
    u'''List hypernyms for word

    >>> hypernyms('fedora')
    ['chapeau', 'hat', 'lid']
    '''
    return sorted(set(lemma.name().replace('_', ' ')
                      for synset in wordnet.synsets(word)
                      for hypernym in synset.hypernyms()
                      for lemma in hypernym.lemmas()
                      if lemma.name().replace('_', ' ') != word))
```


How to use synonyms in i-t-b?

- Select a candidate
- Press a hotkey
- The candidate list is populated with synonyms for the selected candidate

That makes ibus-typing-booster useful, even when one does not need it to speed up typing, i.e. even fast typists can have some benefit.

Part-of-Speech Tagging (POS)

```
>>> text = nltk.word_tokenize("They refuse to permit us to obtain the refuse permit")
>>> text
['They', 'refuse', 'to', 'permit', 'us', 'to', 'obtain', 'the', 'refuse', 'permit']


>>> nltk.pos_tag(text)
[('They', 'PRP'), ('refuse', 'VBP'), ('to', 'TO'), ('permit', 'VB'), ('us', 'PRP'), ('to', 'TO'), ('obtain', 'VB'), ('the', 'DT'), ('refuse', 'NN'), ('permit', 'NN')]
```

→ Quite accurate for English in many cases.

Help for parts of speech

```
>>> nltk.help.upenn_tagset('PRP')
PRP: pronoun, personal
    hers herself him himself hisself it itself me myself one oneself ours
    ourselves ownself self she thee theirs them themselves they thou thy us

>>> nltk.help.upenn_tagset('VBP')
VBP: verb, present tense, not 3rd person singular
    predominate wrap resort sue twist spill cure lengthen brush terminate
    appear tend stray glisten obtain comprise detest tease attract
    emphasize mold postpone sever return wag ...
>>> nltk.help.upenn_tagset('TO')
TO: "to" as preposition or infinitive marker
    to
>>> nltk.help.upenn_tagset('VB')
VB: verb, base form
    ask assemble assess assign assume atone attention avoid bake balkanize
    bank begin behold believe bend benefit bevel beware bless boil bomb
    boost brace break bring broil brush build ...
>>> nltk.help.upenn_tagset('DT')
DT: determiner
    all an another any both del each either every half la many much nary
    neither no some such that the them these this those
>>> nltk.help.upenn_tagset('NN')
NN: noun, common, singular or mass
    common-carrier cabbage knuckle-duster Casino afghan shed thermostat
    investment slide humour falloff slick wind hyena override subhumanity
    machinist ...
```


How to use POS in i-t-b?

- Sometimes there is no trigram data available for the context the user typed
- But the context can still be POS-tagged in that case
- The number of candidates for the next word can then be reduced significantly and/or sorted better because some POS patterns are more likely than others

POS-Tagged Data for Indian languages

Bangla: ক ুঁড়েঘরগ ুল রি/'NN' আকা র/'NN' ব াংলা র/'NNP' ব া/'CC' ভা রতব র্ষে/'NNP' ?/None
ন য/'JJ' ?/None এ ঞ্চলরে/'NN' প রচল তি/'JJ' ক ুঁড়ে/'NN' ঘর/'NN' নয়/'VM' ক্ি/'SYM'
Hindi: पा किस्त ान/'NNP' की/'PREP' प ूर्व/'JJ' प्रधा नम ंत्री/'NN' बेनजी र/'NNPC' भ ुट्टो/'NNP'
पर/'PREP' लगे/'VFM' भ्रष्ट ाच ार/'NN' के/'PREP' आरोपों/'NN' के/'PREP' खिल ाफ/'PREP' भ ुट्टो/'NNP'
द्व ारा/'PREP' द ायर/'NVB' की/'VFM' गई/'VAUX' य ाचिक ा/'NN' की/'PREP' स ुनव ाई/'NN'
म ंगलव ार/'NN' को/'PREP' वकीलों/'NN' की/'PREP' हड़त ाल/'NN' के/'PREP' क ारण/'PREP'
स्थ गित/'JVB' कर/'VFM' दी/'VAUX' गई/'VAUX' ।/'PUNC'
Marathi: ग्र ामीण/'JJ' जिल्ह ाध्यक्ष/'NN' ब ाळ सा हेब/'NNPC' भ ोसले/'NNP' य ांच्य ा/'PRP' ?/None
ध्यक्षतेख ाली/'NN' पक्ष ाची/'NN' आज/'NN' ब ैठ?/None क/'NN' झ ाली/'VM' ./'SYM'
Telugu: ఖ ాదీల/'NN' న ుంచి/'PREP' వచ్చిన/'VJJ' ప త్రాల/'NN' ను/'PREP' స ాక్ష్య ా/'NN'

(From the NLTK Book) So we have POS data not only for English but also for some other languages, although the amount of data seems to be small for many languages

Possible problems

- NLTK is not very fast
- Only for a few dozen languages substantial digital resources are available for use in NLP/NLTK