Part of Speech Tagging

MAS.S60

Catherine Havasi

Rob Speer

Tagsets

- What is a tagset?
- Standards and tagging
- Tags for parts of speech:
  - Nouns, verbs, adverbs, adjectives, articles, etc
  - Subtagging
    - nouns can be singular or plural
    - verbs have tenses
  - Different tagsets have different focuses

Tags are cryptic

```
>>> text = nltk.word_tokenize("And now for something completely different")
>>> nltk.pos_tag(text)
[('And', 'CC'), ('now', 'RB'), ('for', 'IN'), ('something', 'NN'),
('completely', 'RB'), ('different', 'JJ')]
```

- Brown and Treebank established some cryptic tags; everyone tends to use Treebank's
  - CC = coordinating conjunction
  - RB = adverb
  - IN = preposition
  - NN = noun
  - JJ = adjective

NLTK can help

- We don’t even remember what all the tags mean sometimes
- but nltk.help.upenn_tagset(*tag*) does!

Homographs

```
>>> text = nltk.word_tokenize("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')]
```

Tags in NLTK

```
>>> tagged_token = nltk.tag.str2tuple('fly/NN')
>>> tagged_token
('fly', 'NN')
>>> tagged_token[0]
'fly'
>>> tagged_token[1]
'NN'
```

- Tags are tuples
- Tags can be converted by NLTK between tagsets

Making tuples from a corpus

```
>>> sent = '''
... The/AT grand/JJ jury/NN commented/VBD on/IN a/AT
number/NN of/IN
... other/AP topics/NNS ,/, AMONG/IN them/PPO the/AT
Atlanta/NP and/CC
... Fulton/NP-tl County/NN-tl purchasing/VBG departments/
NNS which/WDT it/PPS
... said/VBD ``/`` ARE/BER well/QL operated/VBN and/CC
follow/VB generally/RB
... accepted/VBN practices/NNS which/WDT inure/VB to/IN
the/AT best/JJT
... interest/NN of/IN both/ABX governments/NNS ''/'' ./.
... '''
>>> [nltk.tag.str2tuple(t) for t in sent.split()]
[('The', 'AT'), ('grand', 'JJ'), ('jury', 'NN'),
('commented', 'VBD'),
('on', 'IN'), ('a', 'AT'), ('number', 'NN'), ... ('.', '.')
```

Many Tag Sets

- Different corpora have different conventions for tagging.
- There are ISO standards for tagging…
- There were many boring meetings
- NLTK made a simplified, unified tagset
- … which no one uses.

Simplified Tagset of NLTK

Tag	Meaning	Examples
ADJ	adjective	*new, good, high, special, big, local*
ADV	adverb	*really, already, still, early, now*
CNJ	conjunction	*and, or, but, if, while, although*
DET	determiner	*the, a, some, most, every, no*
EX	existential	*there, there's*
FW	foreign word	*dolce, ersatz, esprit, quo, maitre*
MOD	modal verb	*will, can, would, may, must, should*
N	noun	*year, home, costs, time, education*
NP	proper noun	*Alison, Africa, April, Washington*
NUM	number	*twenty-four, fourth, 1991, 14:24*
PRO	pronoun	*he, their, her, its, my, I, us*
P	preposition	*on, of, at, with, by, into, under*
TO	the word *to*	*to*
UH	interjection	*ah, bang, ha, whee, hmpf, oops*
V	verb	*is, has, get, do, make, see, run*
VD	past tense	*said, took, told, made, asked*
VG	present participle	*making, going, playing, working*
VN	past participle	*given, taken, begun, sung*
WH	*wh* determiner	*who, which, when, what, where, how*

Tagging in Other Languages

```
>>> nltk.corpus.sinica_treebank.tagged_words()
[('\xe4\xb8\x80', 'Neu'), ('\xe5\x8f\x8b\xe6\x83\x85', 'Nad'), ...]
>>> nltk.corpus.indian.tagged_words()
[('\xe0\xa6\xae\xe0\xa6\xb9\xe0\xa6\xbf\xe0\xa6\xb7\xe0\xa7\x87\xe0\xa6\xb0', 'NN'),
('\xe0\xa6\xb8\xe0\xa6\xa8\xe0\xa7\x8d\xe0\xa6\xa4\xe0\xa6\xbe\xe0\xa6\xa8', 'NN'),
...]
>>> nltk.corpus.mac_morpho.tagged_words()
[('Jersei', 'N'), ('atinge', 'V'), ('m\xe9dia', 'N'), ...]
>>> nltk.corpus.conll2002.tagged_words()
[('Sao', 'NC'), ('Paulo', 'VMI'), ('(', 'Fpa'), ...]
>>> nltk.corpus.cess_cat.tagged_words()
[('El', 'da0ms0'), ('Tribunal_Suprem', 'np0000o'), ...]
```

Bangla: ক ুঁড়েঘরগ ুল রি/'NN' আক ার/'NN' ব াংল ার/'NNP' ব া/'CC' ভ ারতব র্ষে/'NNP' ?/None
ন য/'JJ' ?/None এঞ চল রে/'NN' প রচল তি/'JJ' ক ুঁড়ে/'NN' ঘর/'NN' নয়/'VM' ক/'SYM'
Hindi: प ाकिस्त ान/'NNP' की/'PREP' पू र्व/'JJ' प्रध ानम ंत्री/'NN' बेनज ीर/'NNPC' भ ुट्टो/'NNP'
पर/'PREP' लगे/'VFM' भ्रष्ट ाच ार/'NN' के/'PREP' आर ोपों/'NN' के/'PREP' खिल ाफ/'PREP' भ ुट्टो/'NNP'
द्व ार ा/'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' जि ल्ह ाध्यक्ष/'NN' ब ाळ स ाहेब/'NNPC' भ ोसले/'NNP' य ांच्य ा/'PRP' ?/None
ध यक्षतेख ाली/'NN' पक्ष ाची/'NN' आज/'NN' ब ै?/None क/'NN' झ ाली/'VM' ./'SYM'
Telugu: ఖాతాదారుల/'NN' నుంచి/'PREP' వచ్చిన/'VJJ' ప త్రాల/'NN' ను/'PREP' సాక్ష్య ా/'NN'

Finding the Tagset

```
>>> from nltk.corpus import brown
>>> brown_news_tagged =
brown.tagged_words(categories='news',
simplify_tags=True)
>>> tag_fd = nltk.FreqDist(tag for (word, tag) in
brown_news_tagged)
>>> tag_fd.keys()
['N', 'P', 'DET', 'NP', 'V', 'ADJ', ',', '.', 'CNJ',
'PRO', 'ADV', 'VD', ...]
```

Finding the Verbs

```
>>> wsj =
nltk.corpus.treebank.tagged_words(simplify_tags=Tru
e)
>>> word_tag_fd = nltk.FreqDist(wsj)
>>> [word + "/" + tag for (word, tag) in
word_tag_fd if tag.startswith('V')]
['is/V', 'said/VD', 'was/VD', 'are/V', 'be/V',
'has/V', 'have/V', 'says/V', 'were/VD', 'had/VD',
'been/VN', "'s/V", 'do/V', 'say/V', 'make/V', 'did/
VD', 'rose/VD', 'does/V', 'expected/VN', 'buy/V',
'take/V', 'get/V', 'sell/V', 'help/V', 'added/VD',
'including/VG', 'according/VG', 'made/VN', 'pay/
V', ...]
```

Evaluation

- Gold standard: Corpus that has been manually annotated.
- Usually have a training and test component; ideally a portion held out for final evaluation
- Compare algorithms against each other
- Inner Annotator Agreement (IAA)

Data Set Partitioning

```
>>> size = int(len(brown_tagged_sents) * 0.9)
>>> size
4160
>>> size2 = int(len(brown_tagged_sents) * 0.95)
>>> train_sents = brown_tagged_sents[:size]
>>> test_sents = brown_tagged_sents[size:size2]
>>> unigram_tagger = nltk.UnigramTagger(train_sents)
>>> unigram_tagger.evaluate(test_sents)
0.8120203329014252 8
```

Tagging in NLTK

- The “Default” tagger tags each token with the most common tag.

Default Tagger in Action

```
>>> tags = [tag for (word, tag) in
brown.tagged_words(categories='news')]
>>> nltk.FreqDist(tags).max()
'NN'
 >>> raw = 'I do not like green eggs and ham, I do not
 like them Sam I am!'
 >>> tokens = nltk.word_tokenize(raw)
 >>> default_tagger = nltk.DefaultTagger('NN')
 >>> default_tagger.tag(tokens)
 [('I', 'NN'), ('do', 'NN'), ('not', 'NN'), ('like',
 'NN'), ('green', 'NN'),
 ('eggs', 'NN'), ('and', 'NN'), ('ham', 'NN'), (',',
 'NN'), ('I', 'NN'),
 ('do', 'NN'), ('not', 'NN'), ('like', 'NN'), ('them',
 'NN'), ('Sam', 'NN'),
 ('I', 'NN'), ('am', 'NN'), ('!', 'NN')]

 >>> default_tagger.evaluate(brown_tagged_sents)
 0.13089484257215028
```

Regular Expression Tagging

- Uses human-defined patterns:

```
>>> patterns = [
...     (r'.*ing$', 'VBG'),    # gerunds
...     (r'.*ed$', 'VBD'),     # simple past
...     (r'.*es$', 'VBZ'),     # 3rd singular present
...     (r'.*ould$', 'MD'),    # modals
...     (r'.*\'s$', 'NN$'),    # possessive nouns
...     (r'.*s$', 'NNS'),         # plural nouns
...     (r'^-?[0-9]+(.[0-9]+)?$', 'CD'),  # cardinal num.
...     (r'.*', 'NN')          # nouns (default)
... ]
```

RE Tagger in Action

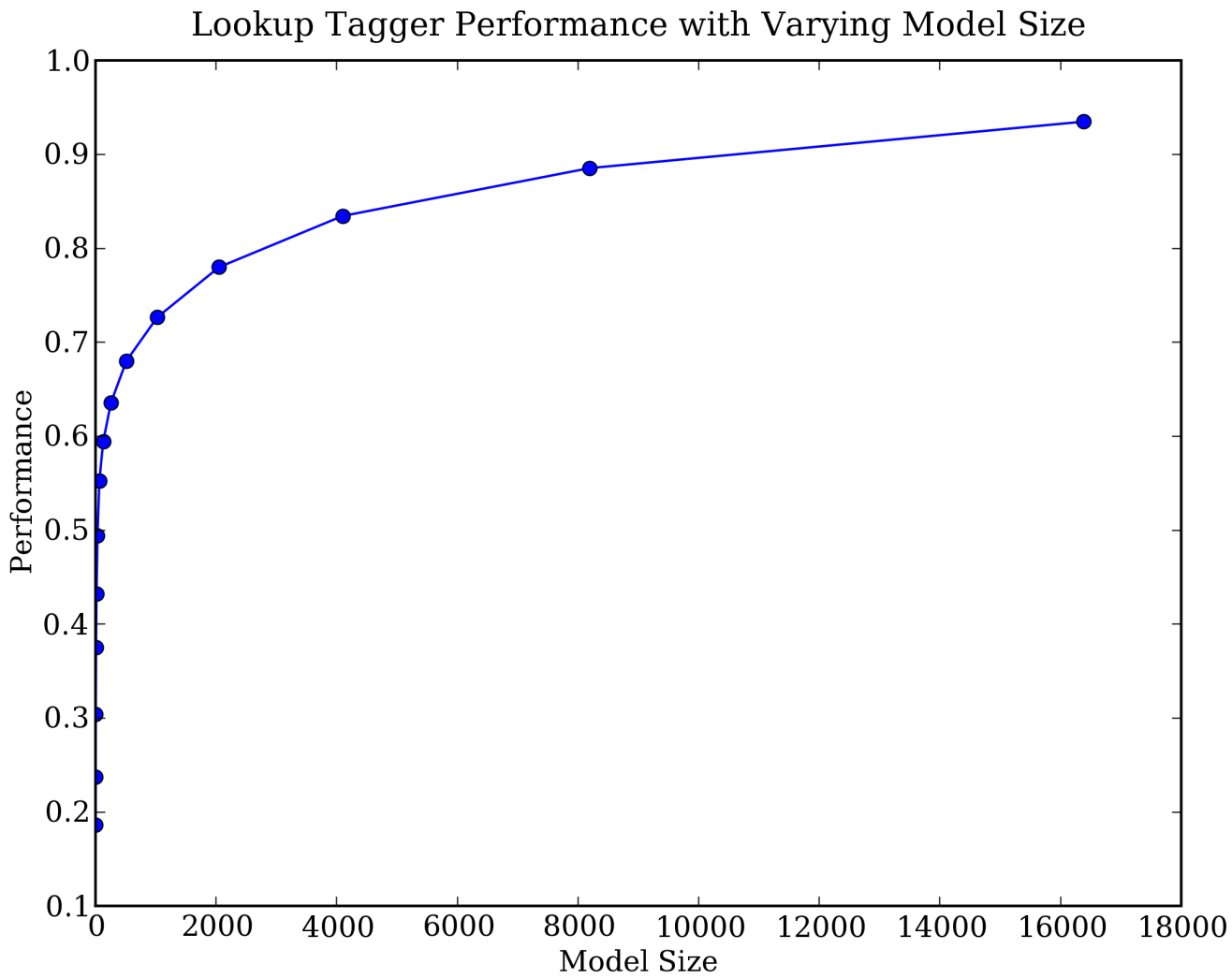
```
>>> regexp_tagger = nltk.RegexpTagger(patterns)
>>> regexp_tagger.tag(brown_sents[3])
[(```, 'NN'), ('Only', 'NN'), ('a', 'NN'), ('relative', 'NN'),
('handful', 'NN'), ('of', 'NN'), ('such', 'NN'), ('reports',
'NNS'), ('was','NNS'), ('received', 'VBD'), ("''", 'NN'), (',',
'NN'), ('the', 'NN'), ('jury', 'NN'), ('said', 'NN'), (',',
'NN'), (```, 'NN'), ('considering', 'VBG'), ('the', 'NN'),
('widespread', 'NN'), ...]
>>> regexp_tagger.evaluate(brown_tagged_sents)
0.20326391789486245
```

Unigram Tagger

- Otherwise known as the “lookup tagger”
- Stores and looks up the tag for each word

```
>>> fd = nltk.FreqDist(brown.words(categories='news'))
>>> cfd =
nltk.ConditionalFreqDist(brown.tagged_words(categories='news'))
>>> most_freq_words = fd.keys()[:100]
>>> likely_tags = dict((word, cfd[word].max()) for word in
most_freq_words)
>>> baseline_tagger = nltk.UnigramTagger(model=likely_tags)
>>> baseline_tagger.evaluate(brown_tagged_sents)
0.4557849513694134
```

Unigram Tagger Performance

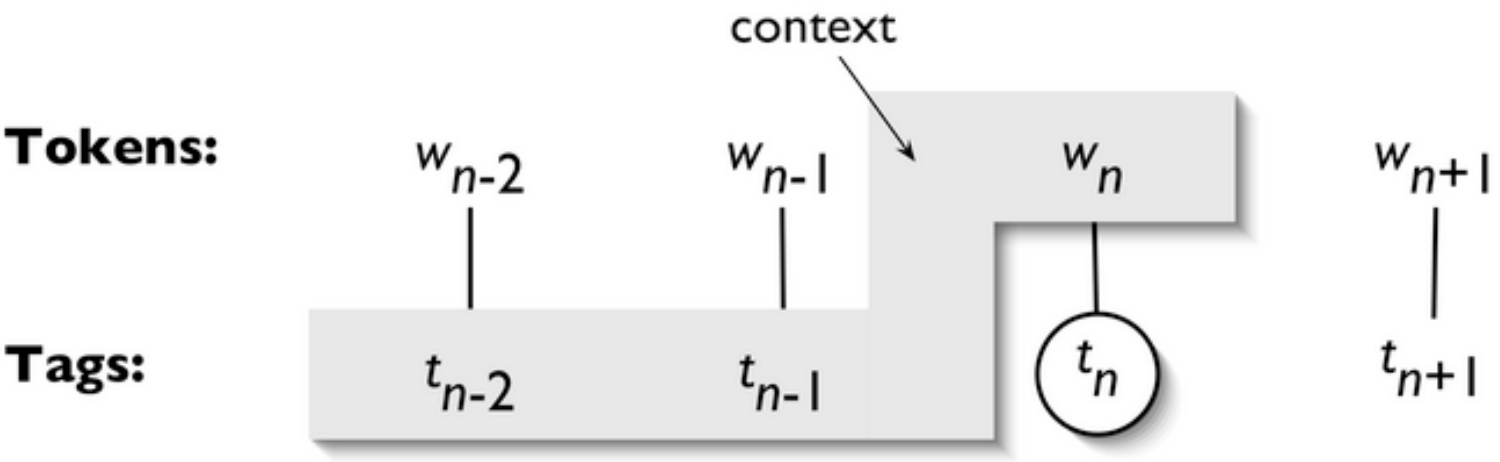


Smaller Data Unigram

- Store less data – only pay attention to non-nouns?
- Saves ~18%
- Does this matter?

General N-Gram Tagging

- Current word plus context: the n-1 previous POS tags.



The Bigram Tagger

```
>>> bigram_tagger = nltk.BigramTagger(train_sents)
>>> bigram_tagger.tag(brown_sents[2007])
[('Various', 'JJ'), ('of', 'IN'), ('the', 'AT'), ('apartments',
'NNS'), ('are', 'BER'), ('of', 'IN'), ('the', 'AT'),
('terrace', 'NN'), ('type', 'NN'), (',', ','), ('being',
'BEG'), ('on', 'IN'), ('the', 'AT'), ('ground', 'NN'),
('floor', 'NN'), ('so', 'CS'), ('that', 'CS'), ('entrance',
'NN'), ('is', 'BEZ'), ('direct', 'JJ'), ('.', '.')]

>>> unseen_sent = brown_sents[4203]
>>> bigram_tagger.tag(unseen_sent)
[('The', 'AT'), ('population', 'NN'), ('of', 'IN'), ('the',
'AT'), ('Congo', 'NP'), ('is', 'BEZ'), ('13.5', None),
('million', None), (',', None), ('divided', None), ('into',
None), ('at', None), ('least', None), ('seven', None),
('major', None), ('``', None), ('culture', None), ('clusters',
None), ("''", None), ('and', None), ('innumerable', None),
('tribes', None), ('speaking', None), ('400', None),
('separate', None), ('dialects', None), ('.', None)]
```

Evaluating the Bigram Tagger

```
>>> bigram_tagger.evaluate(test_sents)
0.10276088906608193
```

Ick.

Why so bad?

- The bigram tagger has no **backoff**
- The unigram and regex taggers and a default assumption…
- Backoff as a baseline (Semeval)

Layering taggers

- Use the benefits of several types of taggers
  - Try the bigram tagger
  - When it is unable to find a tag, use the unigram tagger
  - If that fails, then use the default tagger

Evaluating the Layered Tagger

```
>>> t0 = nltk.DefaultTagger('NN')
>>> t1 = nltk.UnigramTagger(train_sents, backoff=t0)
>>> t2 = nltk.BigramTagger(train_sents, backoff=t1)
>>> t2.evaluate(test_sents)
0.8449117910894049 5
```

That’s awesome, right?

What's wrong with this picture?

- Size of their n-gram table language model
- Context!
  - Words, not just pos, matter in context
- Understandable rules

The Brill Tagger

- A “Transformation based” tagger
- Guess the tag of each word – then go back and fix mistakes
- Compiles a list of correctional rules

Brill Output

```
>>> nltk.tag.brill.demo()
Training Brill tagger on 80 sentences...
Finding initial useful rules...
    Found 6555 useful rules.

           B      |
   S   F   r   O  |          Score = Fixed - Broken
   c   i   o   t  |   R      Fixed = num tags changed incorrect -> correct
   o   x   k   h  |   u      Broken = num tags changed correct -> incorrect
   r   e   e   e  |   l      Other = num tags changed incorrect -> incorrect
   e   d   n   r  |   e
------------------+-------------------------------------------------------
  12  13   1   4  | NN -> VB if the tag of the preceding word is 'TO'
   8   9   1  23  | NN -> VBD if the tag of the following word is 'DT'
   8   8   0   9  | NN -> VBD if the tag of the preceding word is 'NNS'
   6   9   3  16  | NN -> NNP if the tag of words i-2...i-1 is '-NONE-'
   5   8   3   6  | NN -> NNP if the tag of the following word is 'NNP'
   5   6   1   0  | NN -> NNP if the text of words i-2...i-1 is 'like'
   5   5   0   3  | NN -> VBN if the text of the following word is '*-1'
  ...
>>> print(open("errors.out").read())
             left context |    word/test->gold     | right context
--------------------------+------------------------+--------------------------
                          |      Then/NN->RB       | ,/, in/IN the/DT guests/N
, in/IN the/DT guests/NNS |       '/VBD->POS       | honor/NN ,/, the/DT speed
'/POS honor/NN ,/, the/DT |    speedway/JJ->NN     | hauled/VBD out/RP four/CD
NN ,/, the/DT speedway/NN |     hauled/NN->VBD     | out/RP four/CD drivers/NN
DT speedway/NN hauled/VBD |       out/NNP->RP      | four/CD drivers/NNS ,/, c
dway/NN hauled/VBD out/RP |      four/NNP->CD      | drivers/NNS ,/, crews/NNS
hauled/VBD out/RP four/CD |    drivers/NNP->NNS    | ,/, crews/NNS and/CC even
P four/CD drivers/NNS ,/, |     crews/NN->NNS      | and/CC even/RB the/DT off
NNS and/CC even/RB the/DT |    official/NNP->JJ    | Indianapolis/NNP 500/CD a
                          |     After/VBD->IN      | the/DT race/NN ,/, Fortun
ter/IN the/DT race/NN ,/, |    Fortune/IN->NNP     | 500/CD executives/NNS dro
s/NNS drooled/VBD like/IN |  schoolboys/NNP->NNS   | over/IN the/DT cars/NNS a
olboys/NNS over/IN the/DT |       cars/NN->NNS     | and/CC drivers/NNS ./.
```

Evaluating the Brill tagger

- Train over 3500 Treebank sentences
  - Unigram: 89.0%
  - Bigram: 89.7%
  - Brill: 90.1%
- Incremental improvements and NLP publishing

In-class lab: improve the Brill tagger

- The Brill tagging demo is easy to modify
- Look at the error list: you probably see some simple rules that would help
  - Which rules?
  - Where in the chain do you add them?
- Try out taggers with different rules. Who can get the most accuracy?

Taggers are just classifiers

- What the Brill tagger is doing may seem familiar from the classifier lecture
- The modern approach: just turn your training data into features and throw them into a good classifier
- We don't have time to train a **good** classifier, so here's Naïve Bayes again

92% accuracy!

- Imagine what we could do with a better classifier
  - Hidden Markov Models
  - Maximum Entropy decision trees

MaxEnt gets > 99% correct

- ...on the Wall Street Journal
- This is the tagger built in as nltk.pos_tag()

Slide & Graphic Thanks

- Some slides, code, graphics: NLTK & Ed Loper
- Powerpoint of NLTK code: Lillian Cassel