Accessing Lexical Resources

Dr. Jasmeet Singh
TIET

Lexical Resources

- A lexicon, or lexical resource, is a collection of words and/or phrases along with associated information, such as part-of-speech and sense definitions.
- Lexical resources are secondary to texts, and are usually created and enriched with the help of texts.
- Some common lexical resources provided in nltk are:
- Wordlist Corpora
- Stopwords
- Names
- Comparative Wordlists
- Wordnets

Wordlist Corpora

- NLTK includes Word Corpus that includes the wordlsits of the English language.
- We can use it to find unusual or misspelled words in a text corpus.

Example 1

- Filtering a text: This program computes the vocabulary of a text, then removes all items that occur in an existing wordlist, leaving just the uncommon or misspelled words.

```
def incorrect_words(text):
text_vocab = set(w.lower() for w in text if w.isalpha())
english_vocab = set(w.lower() for w in words.words())
unusual = text_vocab.difference(english_vocab)
return sorted(unusual)

incorrect_words(gutenberg.words('austen-emma.txt'))
```

Stopwords

- Stopwords are high-frequency words such as the, to, and also that we sometimes want to filter out of a document before further processing.
- Stopwords usually have little lexical content, and their presence in a text fails to distinguish it from other texts.

Example 2

- A function to compute what fraction of words in a text are not in the stopwords list:

```
def content_fraction(text):
…    stopwords = nltk.corpus.stopwords.words('english')
…    content = [w for w in text if w.lower() not in stopwords]
…    return len(content) / len(text)
...
 content_fraction(gutenberg.words())
```

Example 3

A wordlist is useful for solving word puzzles. How many words of six letters or more can you make from those shown? Each letter must be used once only and must contain center letter.

E	G	I
V	**R**	V
O	N	L

```
freq_letters=nltk.FreqDist('egivrvonl')
Obligatory='r'
wordlist=words.words()
[w for w in wordlist if len(w) >= 6 and obligatory in w and nltk.FreqDist(w)<=freq_letters]
```

Names Corpus

- The Names Corpus, contains 8,000 first names categorized by gender.
- The male and female names are stored in separate files

Example 4

- In the Names Corpus, find the number of female and male names ending with each letter of the alphabet; What do you infer from it

```
cfd = nltk.ConditionalFreqDist(
...        (fileid, name[-1])
...     for fileid in names.fileids()
...     for name in names.words(fileid))
 cfd.plot()
```

- This plot shows that most names ending with a, e, or i are female; names ending in h and l are equally likely to be male or female; names ending in k, o, r, s, and t are likely to be male.

WordNet

- WordNet is a semantically oriented dictionary of English, similar to a traditional thesaurus but with a richer structure.
- NLTK includes the English WordNet, with 155,287 words and 117,659 synonym sets.
- Some methods provided in synsets are:
  - wordnet.synsets('motorcar')[0:] -returns synonym set of word. a synset, or “synonym set,” a collection of synonymous words (or “lemmas”)
  - Word like motorcar can have single synset but words like 'car' can have multiple synsets.
  - Wordnet.synsets('car')[0:]

WordNet Contd…..

Method	Function
wordnet.synsets('car')[0:] Wornet.synsets('motorcar')[0:]	Return synonym set (s) for each word
wordnet.synset('car.n.01').lemma.names()	Return synonym words for the specified synonym sets.
wordnet.synset('car.n.01').definition()	Return definition of words belonging to the specified synonym set.
wordnet.synset('car.n.01').examples()	Return definition of words belonging to the specified synonym set.
wn.synset('car.n.01').lemmas	all the lemmas for a given synset

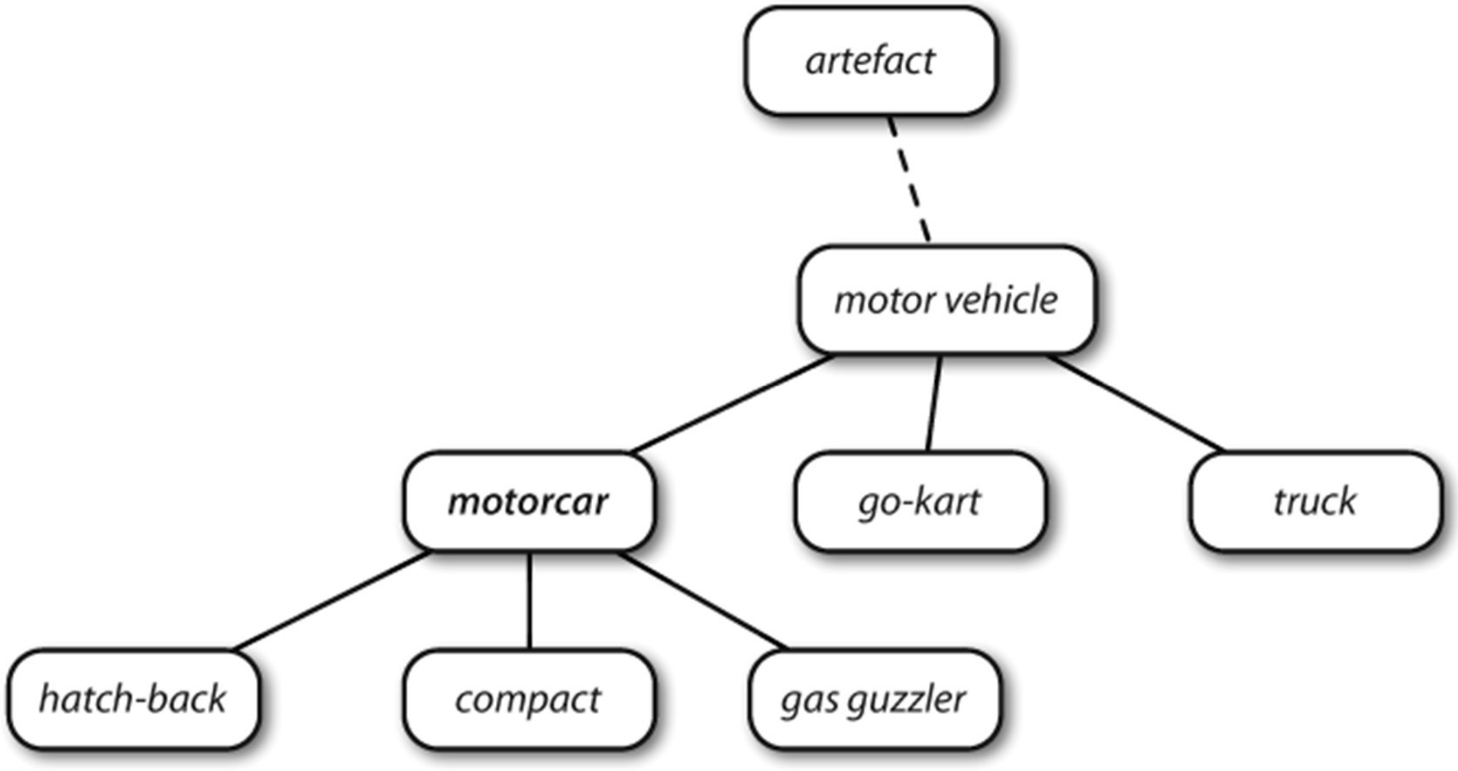
Example 5

- Find the synonym words and definition for each synonym set for the word 'car', 'dog'
- for synset in wordnet.synsets('car'):
- ...   print synset.lemma_names(),synset.definition()
- for synset in wordnet.synsets('dog'):
- ...   print synset.lemma_names(),synset.definition

Wordnet Hierarchy

WordNet synsets correspond to abstract concepts, and they don't always have corresponding words in English.

these concepts are linked together in a hierarchy. Some concepts are very general Others, such as gas guzzler and hatchback, are much more specific.



Wordnet Hierarchy Contd….

hyponyms()	the relation between subordinate	wordnet.synset('car.n.01').hyponyms()
hypernyms()	the relation between superordiante	wordnet.synset('car.n.01').hypernyms()
root_hypernyms()	Root of the synset	wordnet.synset('car.n.01').root_hypernyms()
part_meronyms()`	Parts or the components of the synset	wordnet.synset('tree.n.01').part_meronyms()
substance_meronyms()	Things which the synset is made of	wordnet.synset('tree.n.01').substance_meronyms()

Some other Lexical Relationships

Entailments- There are also relationships between verbs. For example, the act of walking involves the act of stepping, so walking entails stepping. Some verbs have multiple entailments:

```
>>> wn.synset('walk.v.01').entailments()
[Synset('step.v.01')]
>>> wn.synset('eat.v.01').entailments()
[Synset('swallow.v.01'), Synset('chew.v.01')]
```

Relationship between lemmas

Some lexical relationships hold between lemmas, e.g., antonymy:

```
>>> wn.lemma('supply.n.02.supply').antonyms()
[Lemma('demand.n.02.demand')]
>>> wn.lemma('rush.v.01.rush').antonyms()
[Lemma('linger.v.04.linger’)]
```

Derivationally related forms- derivational variants

```
>>> vocal = wn.lemma('vocal.a.01.vocal')
>>> vocal.derivationally_related_forms()
[Lemma('vocalize.v.02.vocalize’)]
```

Pertainyms- related to

```
>>> vocal.pertainyms()
[Lemma('voice.n.02.voice')]
```

Comparative Wordlists

- NLTK includes so-called Swadesh wordlists, lists of about 200 common words in several languages.
- The languages are identified using an ISO 639 two-letter code.
- swadesh.fileids()- returns languages
- swadesh.words('en')

Comparative Wordlists Contd...

- We can access cognate words from multiple languages using the entries() method
- fr2en = swadesh.entries(['fr', 'en'])
- translate = dict(fr2en)
- de2en = swadesh.entries(['de', 'en'])
- es2en = swadesh.entries(['es', 'en'])
- translate.update(dict(de2en))
- translate.update(dict(es2en))

Comparative Wordlists Contd...

- We can compare words in various Germanic and Romance languages:
- languages = ['en', 'de', 'nl', 'es', 'fr', 'pt', 'la']
- for i in [139, 140, 141, 142]:
- … print swadesh.entries(languages)[i]
- …