Natural Language Processing Lab

Week 2 Spell Checker

Using Web Corpus

2017-02-21

Example of Spelling Errors in Context

- Non-word errors (from *Birkbeck Spelling Error Corpus*)
  - I felt very **strang** $\rightarrow$ I felt very **strange**
  - in the **weanter** when it was snowing $\rightarrow$ in the **winter** when it was snowing
- Real word errors
  - at **brake** time $\rightarrow$ at **break** time
  - when the **brack** was finished $\rightarrow$ when the **break** was finished

Correcting Spelling Errors in Context

- Correcting non-word errors
  - Use spell.py by Peter Norvig
  - 1-gram counts (for selecting the best correction)
    - big.txt
    - NetSpeak API
- Correcting real word errors (based on **Bergsma, Lin & Goebel** (2009) *Web-Scale N-gram Models for Lexical Disambiguation*)
  - Use a set of confusable words
  - Replace words in test phrase generating several candidates
  - Conpute overlapping ngram (3-gram) count of candidates (to select the best correction)
    - count('when the break was finished')
      = count('when the break') * count('the break was')
      * count('break was finished')

Norvig's spell.py

```
import re, collections
def words(text): return re.findall('[a-z]+',    text.lower())
def train(features):
  model = collections.defaultdict(lambda: 1)
  for f in features:
    model[f] += 1
  return model

NWORDS = train(words(file('big.txt').read()))

alphabet = 'abcdefghijklmnopqrstuvwxyz'
```

Cont.

```
def edits1(word):
  splits  = [(word[:i], word[i:]) for i in range(len(word) + 1)]

  deletes = [a+b[1:]              for a, b in splits if b]
  transps = [a+b[1]+b[0]+b[2:] for a, b in splits if len(b)>1]
  changes = [a+c+b[1:] for a, b in splits for c in alphabet if b]
  inserts    = [a+c+b  for a, b in splits for c in alphabet]
  return set(deletes + transposes + changes + inserts)
```

Cont.

```
def known(words): return set(w for w in words if w in NWORDS)

def known_edits2(word):
  return set(e2 for e1 in edits1(word) for e2 in edits1(e1) \
                if e2 in NWORDS)

def correct(word):
  candidates = known([word]) or known(edits1(word)) \
                or known_edits2(word) or [word]
  return max(candidates, key=NWORDS.get)
```

Lab for Week 2

- Training and Test data
  - lab2.confusables.txt (`www.alphadictionary.com/articles/confused_words.html`)
  - lab2.test.1.txt (183 errors)
  - lab2.test.2.txt (0 errors)
    - Source: Birkbeck Spelling Error Corpus (`ota.ox.ac.uk/headers/0643.xml`)
- Reusable code
  - spell.py
  - NetSpeakAPI.py
- Evaluation
  - Precision = #hits / #corrections
  - Recall = #hits / #errors
  - FalseAlarm = (#corrections - #hits) / #corrections

Discussion

- Where to find confusable words: dictionaries
  - Laurence Urdang's The Dictionary of Confusable Words
  - Andian Room's Dictionary of Confusable Words
  - Dave Dowling's The Wrong Word Dictionary
- Automatic generation of confusable words
  - Hovermale, Dennis & Mehay (2009). Real-word Spelling Correction for CALL (use CMUDict to generate confusable words)
    `citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.169.6124&rep`
    `rep1&type=pdf`
  - Edit Logs
    * WikEd Error Corpus (WikEd)
      `romang.home.amu.edu.pl/wiked/wiked.html`
    * Language Editing Dataset of Academic Texts (LEDAT)
      `www.vtex.lt/en/ledat.html`

References

- Mays, Eric, Fred J. Damerau and Robert L. Mercer. 1991. Context based spelling correction. Information Processing and Management, 23(5), 517–522.
- Islam, Aminul, and Diana Inkpen. "Real-word spelling correction using Google Web IT 3-grams." Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3. Association for Computational Linguistics, 2009.
- Bergsma, Shane, Dekang Lin, and Randy Goebel. "Web-Scale N-gram Models for Lexical Disambiguation." IJCAI. Vol. 9. 2009.