LING 438/538
Computational Linguistics
Sandiway Fong
Lecture 18: 11/1
Today’s Topics

• Excel Spreadsheet for minimum edit distance
  – a bit more explained

• Stemming and Search
  – Textbook section 3.4
Minimum Edit Distance

- **example**
  - assuming
    - insert = 1
    - delete = 1
    - substitution = 2
    - (or 0 for substituting the same character)

- **recursive formula**
  - incrementally computed from minimum edit distances of shorter strings

\[
P(t|c) = \min \begin{cases} 
\text{distance}(i - 1, j) + \text{ins-cost}(t_i) \\
\text{distance}(i - 1, j - 1) + \text{subst-cost}(s_j, t_i) \\
\text{distance}(i, j - 1) + \text{del-cost}(s_j)
\end{cases}
\]

\[
\text{cost: } 1 + 2 + 2 + 1 + 2 = 8
\]
Minimum Edit Distance

- one formula Microsoft Excel implementation

```
min(C2+1,B3+1,B2+if(C$1=$A3,0,2))
```

$ in a cell reference means don’t change when copied from cell to cell
E.g.
in C$1, 1 stays the same in $A3, A stays the same (not 3)
Minimum Edit Distance

- demo example pairs
  - intention, intent: 3
  - intention, intentional: 2
  - intention, ten: 6
  - intention, ton: 6
  - intention, teen: 7

- min edit distance
  (assuming substitution cost 2)
Stemming

- **Normalization Procedure**
  - inflectional morphology:
    - cities $\Rightarrow$ city, improves/improved $\Rightarrow$ improve
  - derivational morphology:
    - transformation/transformational $\Rightarrow$ transform

- **criterion**
  - preserve meaning (word senses)
    - organization $\Leftrightarrow$ organ

- **primary application**
  - information retrieval (IR)
  - efficacy questioned: Harman (1991)

Google didn’t use stemming
Stemming and Search

- *up until very recently* ...
  - **Word Variations (Stemming)**
    - To provide the most accurate results, Google does not use "stemming" or support "wildcard" searches.
    - In other words, Google searches for exactly the words that you enter in the search box.
    - Searching for "book" or "book*" will not yield "books" or "bookstore". If in doubt, try both forms: "airline" and "airlines," for instance
Stemming and Search

Word Variations (Stemming)

Google now uses stemming technology. Thus, when appropriate, it will search not only for your search terms, but also for words that are similar to some or all of those terms. If you search for "pet lemur dietary needs", Google will also search for "pet lemur diet needs", and other related variations of your terms. Any variants of your terms that were searched for will be highlighted in the snippet of text accompanying each result.

- **Google is more successful than other search engines** in part because it returns “better”, i.e. more relevant, information
  - its algorithm (*a trade secret*) is called **PageRank**
- **SEO (Search Engine Optimization)** is a topic of considerable commercial interest
  - **Goal**: How to get your webpage listed higher by PageRank
  - **Techniques**:
    - e.g. by writing keyword-rich text in your page
    - e.g. by listing morphological variants of keywords
- **Google does not use stemming everywhere**
  - and it does not reveal its algorithm to prevent people “optimizing” their pages
Stemming and Search

- search on:
  - *diet requirements*

- **note**
  - can’t use quotes around phrase
  - blocks stemming

- **Statistics:**
  - diet requirements
  - 625,000 hits (2007)
  - (up from 117,000 in 2004)
  - “dietary requirements” 36,000 hits
Stemming and Search

- search on:
  - dietary requirements

- note
  - can’t use quotes around phrase
  - blocks stemming

- Statistics:
  - dietary requirements
  - 506,000 hits (2007)
  - “dietary requirements” 154,000 hits
Stemming

• **IR-centric view**
  – Applies to open-class lexical items only:
    • *stop-word list*: the, below, being, does
    • exclude: determiners, prepositions, auxiliary verbs

• **not full morphology**
  – prefixes generally excluded
    • *(not meaning preserving)*
    • Examples: asymmetric, undo, encoding
Stemming: Methods

• *use a dictionary* (look-up)
  – OK for English, not for languages with more productive morphology, e.g. Japanese, Turkish

• *write rules*, e.g. Porter Algorithm (Porter, 1980)
  – **Example**:
    • Ends in doubled consonant (not “l”, “s” or “z”), remove last character
      – hopping ⇒ hop
      – hissing ⇒ hiss
Stemming: Methods

• *dictionary approach not enough*
  
    – **Example:** (Porter, 1991)
      
      • routed ⇒ route/rout
        
        – At Waterloo, Napoleon’s forces were *routed*
        
        – The cars were *routed* off the highway

  
  – **notes**
    
    • here, the (inflected) verb form is ambiguous
    
    • preceding word (context) does not disambiguate
Stemming: Errors

- **Understemming**: failure to merge
  - **Example**: adhere/adhesion

- **Overstemming**: incorrect merge
  - **Example**: probe/probable
    - **Claim**: -able irregular suffix, root: probare (Lat.)

- **Mis-stemming**: removing a non-suffix (Porter, 1991)
  - **Example**: reply ⇒ rep
Stemming: Interaction

• *interacts with noun compounding*
  – **Example:**
    • *operating systems*
    • *negative polarity items*

  – for IR, compounds need to be identified first…
Stemming: Porter Algorithm

- The Porter Stemmer (Porter, 1980)
- **textbook**
  - *section 3.4*
- **URL**
  - [http://www.tartarus.org/~martin/PorterStemmer/](http://www.tartarus.org/~martin/PorterStemmer/)
- **C code**
  - for English
  - most widely used system
  - manually written rules
  - **5 stage approach** to extracting roots
  - considers suffixes only
  - may produce non-word roots
Stemming: Porter Algorithm

• **rule format:**
  – \((\text{condition on stem}) \ \text{suffix}_1 \Rightarrow \text{suffix}_2\)
    • *In case of conflict, prefer longest suffix match*

• *“Measure”* of a word is \(m\) in:
  – \((\text{C})(\text{VC})^m(\text{V})\)
    • \(\text{C} = \text{sequence of one or more consonants}\)
    • \(\text{V} = \text{sequence of one or more vowels}\)
  – **examples:**
    • *tree* \(\text{C(VC)}^0\text{V} \ \text{(tr)(ee)}\)
    • *troubles* \(\text{C(VC)}^2 \ \text{(tr)(ou)(bl)(e)(s)}\)
Stemming: Porter Algorithm

• Step 1a: *remove plural suffixation*
  – SSES $\Rightarrow$ SS (caresses)
  – IES $\Rightarrow$ I (ponies)
  – SS $\Rightarrow$ SS (caress)
  – S $\Rightarrow$ (cats)

• Step 1b: *remove verbal inflection*
  – $(m>0)$ EED $\Rightarrow$ EE (agreed, feed)
  – (*v*) ED $\Rightarrow$ (plastered, bled)
  – (*v*) ING $\Rightarrow$ (motoring, sing)
Stemming: Porter Algorithm

- Step 1b: *(contd. for -ed and -ing rules)*
  - AT $\Rightarrow$ ATE (conflated)
  - BL $\Rightarrow$ BLE (troubled)
  - IZ $\Rightarrow$ IZE (sized)
  - $(*\text{doubled c} \& \neg(*L \lor *S \lor *Z)) \Rightarrow$ single c (hopping, hissing, falling, fizzing)
  - (m=1 $\& *\text{cvc}) \Rightarrow$ E (filing, failing, slowing)

- Step 1c: Y and I
  - $(*\text{v}*) Y \Rightarrow I$ (happy, sky)
Stemming: Porter Algorithm

• Step 2: *Peel one suffix off for multiple suffixes*
  - (m>0) ATIONAL ⇒ ATE (relational)
  - (m>0) TIONAL ⇒ TION (conditional, rational)
  - (m>0) ENCI ⇒ ENCE (valenci)
  - (m>0) ANCI ⇒ ANCE (hesitanci)
  - (m>0) IZER ⇒ IZE (digitizer)
  - (m>0) ABLI ⇒ ABLE (conformabli) - *able* (step 4)
  - …
  - (m>0) IZATION ⇒ IZE (vietnamization)
  - (m>0) ATION ⇒ ATE (predication)
  - …
  - (m>0) IVITI ⇒ IVE (sensitiviti)
Stemming: Porter Algorithm

• Step 3
  - (m>0) ICATE ⇒ IC (triplicate)
  - (m>0) ATIVE ⇒ (formative)
  - (m>0) ALIZE ⇒ AL (formalize)
  - (m>0) ICITI ⇒ IC (electriciti)
  - (m>0) ICAL ⇒ IC (electrical, chemical)
  - (m>0) FUL ⇒ (hopeful)
  - (m>0) NESS ⇒ (goodness)
Stemming: Porter Algorithm

• Step 4: Delete last suffix
  – (m>1) AL ⇒ (revival) - revive, see step 5
  – (m>1) ANCE ⇒ (allowance, dance)
  – (m>1) ENCE ⇒ (inference, fence)
  – (m>1) ER ⇒ (airliner, employer)
  – (m>1) IC ⇒ (gyroscopic, electric)
  – (m>1) ABLE ⇒ (adjustable, mov(e)able)
  – (m>1) IBLE ⇒ (defensible, bible)
  – (m>1) ANT ⇒ (irritant, ant)
  – (m>1) EMENT ⇒ (replacement)
  – (m>1) MENT ⇒ (adjustment)
  – ...

Stemming: Porter Algorithm

- **Step 5a: remove e**
  - $(m>1) \ E \Rightarrow (\text{probate}, \ \text{rate})$
  - $(m>1 & \neg*\text{cvc}) \ E \Rightarrow (\text{cease})$

- **Step 5b: ll reduction**
  - $(m>1 & *\text{LL}) \Rightarrow L \ (\text{controller}, \ \text{roll})$
Stemming: Porter Algorithm

- **Misses** (understemming)
  - Unaffected:
    - *agreement* \((VC)^1VCC\) - step 4 \((m>1)\)
    - *adhesion*
  - Irregular morphology:
    - drove, geese

- **Overstemming**
  - *relativity* - step 2

- **Mis-stemming**
  - *wander* \(C(VC)^1VC\)
Homework

• with a Google theme...

• due next Thursday
Britney Spears

- http://news.bbc.co.uk/cbbcnews/hi/music/newsid_1953000/1953614.stm
Making Money from Misspellings

http://news.bbc.co.uk/1/hi/sci/tech/1575060.stm

By BBC News Online technology correspondent Mark Ward

US legal authorities are appealing for help in tracking down John Zuccarini, who they say is making more than a million dollars a year from a collection of misspelled domain names.
Making Money from Misspellings

• Excerpts:
  – US legal authorities are appealing for help in tracking down John Zuccarini, who they say is making **more than a million dollars a year** from a collection of misspelled domain names.
  – The Federal Trade Commission is now looking for ways to recover the cash Mr Zuccarini has made from the domain names.

• Excerpts:
  – Mr Zuccarini has been practising a novel variation of cybersquatting which usually involves gaining control of a website that you have no real claim to, and then offering it for sale to the rightful owner at a premium.
  – The domains registered by Mr Zuccarini were typically misspellings of well-known names. Mr Zuccarini has reportedly registered **15 variations of the spelling of Cartoon Network TV** channel, and **41 of pop star Britney Spears**.
Making Money from Misspellings

- **So what happened to him?**

- **John Zuccarini** (born c. 1947) is an American businessman who served time in federal prison for violating the **Truth in Domain Names Act**.

- Zuccarini operated a domain name speculation business. He is reported as owning 5500 domains before his arrest.
use this list for the homework
http://www.google.com/jobs/britney.html

1. 488941 britney spears
2. 40134 brittany spears
3. 36315 brittney spears
4. 24342 britany spears
5. 7331 britny spears
6. 6633 briteny spears
7. 2696 britteny spears
8. 1807 briney spears
9. 1635 brittny spears
10. 1479 brintey spears
11. 1479 britanny spears
12. 1338 britiny spears
13. 1211 britnet spears
14. 1096 britiney spears

from the BBC website
typos: #2 is wrong,
#3 and #5 shouldn’t be the same
Edit Distance Formulation

• Come up with an edit distance metric that accounts for the cost of misspelling Britney
• Make your algorithm return costs that mimic the order of the misspellings observed by Google for the top 8 misspellings:
  i.e. cost(brittany) < cost(brittney) < cost(brittney)
• Implement your algorithm in Excel and submit your spreadsheet
• Hint: you could assign different costs for adding a t (vs. any other letter) or for changing e to a etc.
• In Excel you might want to use 26 x 26 tables to look up cost values

Extra Credit: getting top 13 in order

1. 488941 britney spears
2. 40134 brittany spears
3. 36315 brittney spears
4. 24342 britany spears
5. 7331 britny spears
6. 6633 britenyn spears
7. 2696 britteny spears
8. 1807 briney spears
9. 1635 brittny spears
10. 1479 brintey spears
11. 1479 britanny spears
12. 1338 britiny spears
13. 1211 britnet spears
14. 1096 britiney spears