Porter Stemmer

Miriam Butt
October 2003

Background

Stemming is potentially of use for many applications:

- Information Retrieval (indices, e.g.,Web, abstracts)
- Machine Translation (quick way to get a morphology)

Famous Algorithm: Porter Stemmer (Porter 1980)

http://www.tartarus.org/~martin/PorterStemmer/

http://snowball.tartarus.org/

Output

Sample Output (English):

consigned	consign	knack	knack
consignment	consign	knackeries	knackeri
consolation	consol	knaves	knavish
consolatory	consolatori	knavish	knavish
consolidate	consolid	knif	knif
consolidating	consolid	knife	knife
consoling	consol	knew	knew

Output

Sample Output (German):

aufeinander	aufeinand	kategorie	kategori
auferlegen	auferleg	kategorien	kategori
auferlegt	auferlegt	kater	kat
auferlegten	auferlegt	katers	kat
auferstanden	auferstand	katze	katz
auferstehen	auferstand	katzen	katz
aufersteht	aufersteht	kätzchen	katzch

Efficiency

Algorithmic stemmers can be fast (and lean):

E.g.: 1 Million words in 6 seconds on 500 MHz PC

- It is more efficient not to use a dictionary (don't have to maintain it if things change).
- It is better to ignore irregular forms (exceptions) than to complicate the algorithm (not much lost in practice).

Algorithmic Method

Porter Stemmers use simple algorithms to determine which affixes to strip in which order and when to apply repair strategies.

Input	**Strip -ed Affix**	**Repair**
hoped	hop	hope (add -e if word is short)
hopped	hopp	hop (delete one if doubled)

Samples of the algorithms are accessible via the Web and can be programmed in any language.

Advantage: easy to see understand, easy to implement.

Basic Morphology

Basic Affix Typology (don't seem to need more):

- **i-suffix**: inflectional suffix

  English: *cheer+ed = cheered, fit+ed = fitted, love+ed = loved*

- **d-suffix**: derivational suffix, changes word type

  English: *walk(V)+er = walker(N), happy(A)+ness=happiness(N)*

- **a-suffix**: attached suffix (enclitics).

  Italian *mandargli= mandare+gli* = to send + to him

Algorithmic Method

General Strategy:

- Normal order of suffixes seems to be *d, i, a.*
- Remove from right in order *a, i, d.*
- Generally remove all the *a* and *i* suffixes, sometimes leave the *d* one.

Types of Errors

• Conflation:	reply, rep.	rep
• Overstemming:	wander	wand
	news	new
• Misstemming:	relativity	relative
• Understemming:	knavish	knavish

Algorithmic Method

Strategy for German:

- Leave prefixes alone because they can change meaning.
- Put everything in small caps.
- Get rid of *ge-*.
- Get rid of *i* type: *e, em, en, ern, er, es, s, est,* (e.g, *armes* > *arm*)
- Get rid of *d* type*: end, ung, ig, ik, isch, lich, heit, keit*

Information Retrieval

Does stemming indeed improve IR?

- **No:** Harman (1991), Krovetz (1993)
- **Possibly:** Krovetz (1995)
  Depends on type of text, and the assumption is that once one moves beyond English, the difference will prove significant.

Crosslinguistic Applicability

- Can this type of stemming be applied to all languages?
  — Not to Chinese, for example (doesn't need it).
- Do all languages have the same kind of morphology?
  — **No.** Stemming assumes basically agglutinative morphology. This is not true crosslinguistically (but the algorithms seem to work pretty well within Indo-European).

• Porter notes that Old English can be stemmed quite easily using the modern Stemmer, just a few forms need to be respelled, e.g., *-ick* for *-ic*.