Morphology induction

July 23, 2007

Sharon Goldwater

Recap

- Topics covered so far:
  - Word segmentation, syntax, discourse, maximum-likelihood, EM, EM, EM.
- Today:
  - Morphology
  - When not to use EM (and how).

Morphology: what is it?

Input: escape, escapes, escaping, inescapable, possible, impossible, impossibility, reducible, reproducible

Output?

escape escape.s escap.ing in.escap.able possible im.possible im.possib.ility re.ducible re.produc.ible	escape escape+s escape+ing in+escape+ble possible in+possible in+possible+ity re+ducible re+produce+ble	escape escape+3SG escape+PROG NEG+escape+BLE possible NEG+possible NEG+possible+BLE RE+ducible RE+produce+BLE

Possible goals

- Single or multiple suffixes
  - walk, walk.s, walk.er.s
- Prefix/suffix/circumfix
  - re.do.ing, German ge.arbeite.t [work+PERF]
- Fusional
  - Russian chita.et [read+3sg], chita.yut [3pl], chita.ete [2pl]
- Agglutinative
  - Finnish arvo.n.lisä.vero.ttoma.sta [from something exclusive of VAT] (Creutz & Lagus, 2005)
- Ablaut
  - goose/geese, dig/dug, German graben/grub [dig]
- Templatic
  - Hebrew shomer [guard+m_pres_sg], shomrim [m_pres_pl], shamra [f_sg_past]

Sources of information

- Many sources of information are potentially useful.
  - Orthography/phonology
  - Syntax
  - Semantics
  - Frequency

Yarowsky & Wicentowski (2000)

- A “minimally-supervised” alignment algorithm.
  - Input: easily available resources.
  - Output: alignments between morphological variants of each word (including irregulars).
- Example of combining many sources of information to achieve excellent results.

Input

- Corpus
- Dictionary listing POS for roots (may be noisy)
- List of consonants and vowels
- Table of canonical suffixes/POS tags:

English :

Part of Speech	VB	VBD	VBZ	VBG	VBN
Canonical Suffixes	+ϵ	+ed (+t) +ϵ	+s	+ing	+en +ed (+t) +ϵ
Examples (*not* used in training)	jump announce take	jumped announced took	jumps announces takes	jumping announcing taking	jumped announced taken

Spanish:

Part of Speech	VRoot	VPI1s	VPI2s	VPI3s	VPI1p	VPI2p	VPI3p
Canonical Suffixes	+ar +er +ir	+o	+as +es	+a +e	+amos +emos +imos	+áis +éis +ís	+an +en

Table from Y&W (2000)

Output

- List of roots and inflected forms, with morphophonological analysis and POS:

ROOT	STEM CHANGE	SUFFIX	INFLECTION	POS
take	ake → ook	+ϵ	took	VBD
take	e → ϵ	+ing	taking	VBG
take	ϵ → ϵ	+s	takes	VBZ
take	e → ϵ	+en	taken	VBN
skip	ϵ → p	+ed	skipped	VBD
defy	y → i	+ed	defied	VBD
defy	y → ie	+s	defies	VBZ
defy	ϵ → ϵ	+ing	defying	VBG
jugar	gar → eg	+a	juega	VPI3S
jugar	gar → eg	+an	juegan	VPI3P
jugar	ar → ϵ	+amos	jugamos	VPI1P
tener	ener → ien	+en	tienen	VPI3P

Table 1: Target output (English and Spanish)

Table from Y&W (2000)

Method

- Bootstrapping
  - Create several weak learners using complementary sources of information.
    - Each produces a ranked list of related pairs, with confidence scores.
  - Combine individual decisions to get better decisions.
  - Retrain individual components on output.
  - Repeat to convergence.

Sources of information

1. Frequency similarity
   - Inflected forms have similar frequencies:

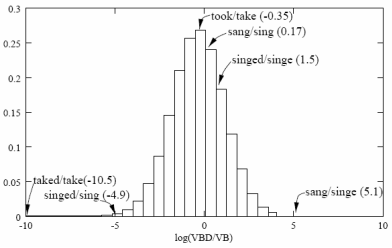


Figure from Y&W (2000)

Sources of information

2. Context similarity
   - Inflected forms have similar arguments: *read the book*, *reading a book*.
   - To avoid confusion from function words, use regular expressions:

     CW-subj (AUX|NEG)* Verb DET? CW* CW-obj

Sources of information

3. Orthographic similarity
   - Inflected forms have similar spellings.
   - Use weighted edit distance to compute similarity, and retrain weights.

Sources of information

4. Morphological rule probabilities
   - How often does each rule occur with each stem-final context, suffix, POS?

Root Context	Stem Change	Suffix	Count	Matching Examples
..ray	$\epsilon \rightarrow \epsilon$	+ed	5	spray, stray,...
...ay	$\epsilon \rightarrow \epsilon$	+ed	13	play, spray,...
...oy	$\epsilon \rightarrow \epsilon$	+ed	3	annoy, enjoy,...
...ey	$\epsilon \rightarrow \epsilon$	+ed	5	obey, key,...
...fy	y $\rightarrow$ i	+ed	21	beautify,...
...ry	y $\rightarrow$ i	+ed	7	carry,...
...dy	y $\rightarrow$ i	+ed	4	bloody,...
...y	y $\rightarrow$ i	+ed	43	carry,...
...y	$\epsilon \rightarrow \epsilon$	+ed	21	spray,...
...y	$\epsilon \rightarrow \epsilon$	+ing	83	carry, spray,...
...e	e $\rightarrow \epsilon$	+ed	728	dance,...
...e	e $\rightarrow \epsilon$	+ing	783	dance, take,...
...e	$\epsilon \rightarrow \epsilon$	+ing	1	singe

Table from Y&W (2000)

Parameter re-estimation

- Update alignment pairs by combining individual weighted lists.
- New list of pairs is used to re-estimate
  - frequency ratios between inflected forms.
  - weights for edit distance measure.
  - probabilities for morphological rules.

Results

- Evaluated on English verbs:

Combination of Similarity Models	# of Iter-ations	All Words (3888)	Highly Irregular (128)	Simple Concat. (1877)	Non-Concat. (1883)
FS (*Frequency Sim*)	(Iter 1)	9.8	18.6	8.8	10.1
LS (*Levenshtein Sim*)	(Iter 1)	31.3	19.6	20.0	34.4
CS (*Context Sim*)	(Iter 1)	28.0	32.8	30.0	25.8
CS+FS	(Iter 1)	32.5	64.8	32.0	30.7
CS+FS+LS	(Iter 1)	71.6	76.5	71.1	71.9
CS+FS+LS+MS	(Iter 1)	96.5	74.0	97.3	97.4
CS+FS+LS+MS	(Convg)	**99.2**	**80.4**	**99.9**	**99.7**

  - Right: knew/know, made/make, brought/bring
  - Wrong: got/go, slew/slit, went/want

Schone & Jurafsky (2001)

- Combining multiple sources of information in a completely unsupervised way.
- Input: text corpus.
- Output: "conflation sets" of related words:

  {abuse, abusive, abusing, abuses, abusers, abused}

Sources of information

- Orthography
  - Initially identify possible related word pairs with different prefix/suffix.
- Semantics
  - Use LSA to compute semantic vectors for words.
  - Estimate whether semantic correlation between pairs is significantly greater than chance.

Sources of information

- Syntax
  - Compute local context vectors for words.
  - Estimate whether syntactic correlation between pairs is significantly different from chance.
- Frequency
  - Eliminate pair relations that are too infrequent.
- Transitive closure
  - Pair is related if there is a path between words.

Results, discussion

- Tested on English, German, Dutch.
  - Performs better than Linguistica (Goldsmith, 2001)
  - Each source of information improves scores.
- Weaknesses:
  - Lots of free parameters!
  - Heuristic search, no model.

Related work

- Wicentowski (2004)
  - Extends morphological rule structure and evaluates on 32 langs (incl. Icelandic, Hindi, Estonian, Klingon).
- Monson (2007), Dasgupta & Ng (2007), etc.
  - Other procedural methods for morphology induction.
- Yarowsky (1995)
  - Semi-supervised bootstrapping algorithm.
- Blum & Mitchell (1998)
  - Co-training.
- Abney (2004)
  - Mathematical analysis of Yarowsky algorithm.

Summary

- Advantages of procedural methods for morphology induction:
  - Can combine many sources of information.
  - Often achieve good performance.
- Disadvantages:
  - No probabilistic model.
    - What is being optimized?
    - How to combine into larger systems?

Model-based induction

- Thought experiment: use maximum-likelihood estimation to learn morphology.
  - Generative model:
    - generate morphological class c
    - generate stem t conditioned on class
    - generate suffix f conditioned on class

$$P(c,t,f) = P(c)P(t|c)P(f|c)$$

- What happens?

Possible results

Input: {lift, lifts, lifting, jump, jumps, jumping, jumped, roll, rolling, rolled, walk, walks, walking, walked, ...}

Output:

$$\left\{\begin{matrix}\text{lift}\\ \text{jump}\\ \text{roll}\\ \text{walk}\\ \text{...}\end{matrix}\right\} \times \left\{\begin{matrix}\epsilon\\ \text{-s}\\ \text{-ed}\\ \text{-ing}\end{matrix}\right\} \quad \text{vs.} \quad \left\{\begin{matrix}\text{lift} & \text{roll}\\ \text{lifts} & \text{rolling}\\ \text{lifting} & \text{rolled}\\ \text{jump} & \text{walk}\\ \text{jumps} & \text{walks}\\ \text{jumping} & \text{walking}\\ \text{jump} & \text{walked}\\ \text{...}\end{matrix}\right\} \times \{\epsilon\}$$

A solution?

- ML doesn't work for comparing models of different sizes (numbers of parameters)
- Possible solution: model selection.
  - Compute ML solution for various different sized models, use held-out data (*) to determine if adding parameters really helped.
  - Examples: Petrov & Klein (2006; 2007)
- Is this always feasible?

Another solution

- Introduce a prior:

$$P(\theta \mid d) \propto P(d \mid \theta)P(\theta)$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, P(d \mid \theta)P(\theta)$$

- What sort of prior should we use?

Obligatory Chomsky quote

In careful descriptive work, we almost always find that one of the considerations involved in choosing among alternative analyses is the simplicity of the resulting grammar. If we can set up elements in such a way that very few rules need be given about their distribution, or that these rules are very similar to the rules for other elements, this fact certainly seems to be a valid support for the analysis in question. It seems reasonable, then, to inquire into the possibility of defining linguistic notions in the general theory partly in terms of such properties of grammar as simplicity. (pp. 113-114)

…

It is tempting, then, to consider the possibility of devising a notational system which converts considerations of simplicity into considerations of length…More generally, simplicity might be determined as a weighted function of the number of symbols, the weighting devised so as to favor reductions in certain parts of the grammar. (p. 117)

(Chomsky, 1955) [emphasis mine]

Minimum description length

(Rissanen, 1989)

- Derived from information theory:
  - Need to encode the corpus so that
    - Codebook (grammar) is short.
    - Encoded corpus is also short.
  - For codebook h and corpus d, minimize
    Length(h) + Length(encoding$_h$(d))

Corpus: walk walks jump jumps jumped eats

Codebook 1:

walk	0
walks	10
jump	110
jumps	1110
jumped	11110
eats	111110

Codebook 2:

walk	0
jump	10
eat	110
s	1110
ed	11110

Codebook 3:

walk	110
jump	0
eat	1110
s	10
ed	11110

Encoding 1: 010110111011110111110

Encoding 2: 00111010101110101111101101110

Encoding 3: 1101101000100111101110

Information theory

- If we choose codes for each item optimally, the length of the encoded corpus will be $-\log_2 P(d|h)$.
- We want to find

$$\begin{aligned}\hat{h} &= \underset{h}{\operatorname{argmin}}\,(\text{len}(e_h(d)) + \text{len}(h)) \\ &= \underset{h}{\operatorname{argmin}}\,(-\log_2 P(d \mid h) + \text{len}(h)) \\ &= \underset{h}{\operatorname{argmax}}\, P(d \mid h) \cdot 2^{-\text{len}(h)} \\ &= \underset{h}{\operatorname{argmax}}\, P(d \mid h) \cdot P(h)\end{aligned}$$

where $P(h)$ increases exponentially with length.

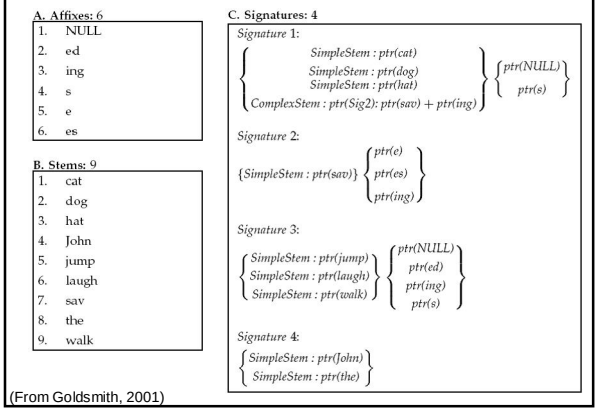
Goldsmith (2001)

- Organizes grammar into signatures:

g_1 = { jump, walk, laugh, saving } x { NULL, ed, s, ing }

g_2 = { sav, lik, escap } x { e, ed, es, ing }

g_3 = { cat, dog } x { NULL, s }



(From Goldsmith, 2001)

Form of codebook

1. List number of stems, suffixes, signatures.
2. List stems and their codes.
3. List suffixes and their codes.
4. List signatures and their codes.
   - Each signature: list of stem codes and suffix codes.

Computing the length of the codebook:

(i) $\lambda\langle T\rangle + \lambda\langle F\rangle + \lambda\langle \Sigma\rangle$

(ii) *Suffix list*: $\sum_{f \in Suffixes} \left(\log 26^{*} length(f) + \log \frac{[W_A]}{[f]} \right)$

(iii) *Stem list*: $\sum_{t \in T} \left(\log 26^{*} length(t) + \log \left(\frac{[W]}{[t]} \right) \right)$

(iv) *Signature component*

Stated once for the whole component:

(a) Signature list: $\sum_{\sigma \in Signatures} \log \frac{[W]}{[\sigma]}$

For *each* signature:

(b) Size of the count of the number of stems plus size of the count of the number of suffixes:

(From Goldsmith, 2001)

…and so on.

Length of encoding

- Encode each word as (*g*,*t*,*f*). Then

$$\begin{aligned} \text{len}(e_h(w)) &= \text{len}(e_h(g_w)) + \text{len}(e_h(t_w)) + \text{len}(e_h(f_w)) \\ &= -\log(P(g_w)P(t_w \mid g_w)P(f_w \mid g_w)) \end{aligned}$$

$$\text{len}(e_h(d)) = -\log \prod_w P(g_w)P(t_w \mid g_w)(f_w \mid g_w)$$

Search

- How to minimize objective function?
  1. Heuristic splits to create initial signatures.
  2. Filter infrequent signatures.
  3. Consider various possible perturbations.
     - Shift a character from each suffix onto stem: {sea,ligh,loo} x {ted.ts.ting}
     - Split compound suffixes: ments -> ment.s
     - Etc.
- Custom search procedure.

Remaining issues

- Orthographic rules.
  {sav,lik} x {e.ed.ing.es} vs.
  {walk,jump} x {NULL.ed.ing.s}
- Profusion of signatures.
  - {NULL,ed,ing,s}, {NULL,ing,s}, {ed,ing,s}, {NULL,ed,er,ing,s}, etc.
- Less effective on agglutinative, other morphology.

Creutz & Lagus (2005; 2007)

- Designed for agglutinative morphology (e.g., Finnish, Turkish)
  - Words split into multiple morphs.
  - Model morphotactics using HMM:
    - Hidden classes: stem, suffix, prefix, none-of-above
  - Search is iterative:
    1. Find initial segmentation (uses older model).
    2. Find morphs to split and/or join.
    3. Resegment and re-estimate parameters with EM.
    4. Repeat 2-3.

Related work

- Other Goldsmith papers
  - Beginnings of: orthographic rules, syntactic context, agglutinative morphology.
- Other MDL
  - Phonology (Ellison, 1994), word segmentation (de Marcken, 1995; Cartwright & Brent, 1996), syntax (Dowman, 2000).

Summary

- Advantages of MDL:
  - Can define (more or less) arbitrary priors.
  - Provides fair comparison between models with different structures and parameters.
- Disadvantages:
  - Uninteresting choices in grammar definition may have major and non-obvious results (see Goldwater & Johnson, 2003).
  - Requires specialized search procedures.
    - Results may be partly due to search, not MDL.