

Minimally Supervised Morphological Analysis by Multimodal Alignment

by David Yarowski and Richard Wicentowski (ACL 2000)

Presented by Ted Meeds and Ulrich Germann

David Yarowsky

- Currently professor of CS at JHU.
- PhD University of Pennsylvania, 1996
- Research Interests: “natural language processing and spoken language systems, machine translation, information retrieval, very large text databases and machine learning”
(from <http://www.cs.jhu.edu/~yarowsky/>)
- Wicentowski's thesis supervisor.

Richard Wicentowski

- Currently Professor of Computer Science at Swarthmore College.
- PhD John Hopkins University, 2002
- Research Interests: computational morphology; word sense disambiguation.
- Work presented today is part of his PhD work.

What the paper is about...

... an algorithm to build morphological analyzers with minimal supervision.

Given

- an **unannotated corpus** in some language
- **a-priori knowledge** of the morphology of that language

Perform the following:

- Build a probabilistic alignment of the *word forms* of that language.
- Use that alignment to learn (with supervision) a morphological analyzer.

In other words, build a table like this:

ROOT	STEM		INFLECTION	POS
	CHANGE	SUFFIX		
take	ake → ook	+ε	took	VBD
take	e → ε	+ing	taking	VBG
take	ε → ε	+s	takes	VBZ
take	e → ε	+en	taken	VCN
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)

from: Yarowsky & Wicentowski (2000)

Why Do We Care?

Morphological analysis

- reduces vocabulary size, which means
 - fewer dimensions in vector space models;
 - higher data density (more counts of fewer event types) for model training.
- facilitates query expansion for IR.
- is necessary for dictionary lookup and indexing.
- ...

Morphological Typology

- Isolating (analytic) languages.
- Agglutinative languages.
- Inflecting languages.
- Incorporating (polysynthetic) languages.

-
- Classification goes back to Schlegel (1818) and Humboldt (1836).
 - Natural Languages rarely fall exclusively into any one single category.

Isolating Languages

- Grammatical functions are expressed by independent units (words).
- Rule of thumb:

$$\frac{\text{\# of morphemes}}{\text{\# of words}} \approx 1.$$

- Prototypical examples: Vietnamese, Classical Chinese.

Agglutinative Languages

- Grammatical functions expressed by affixation.
- Rule of thumb: one grammatical or morphological feature per morpheme.

E.g. Japanese *hataraki.ta.gar.ana.kat.ta=kara*:

<i>hataraki</i>		<i>ta</i>		<i>gar</i>		<i>ana</i>		<i>kat</i>		<i>ta</i>		= <i>kara</i>
“work”		“want to”		[apparently]		“not”		[morphol. ‘glue’]		[PERF]		“because”
“Because [he/she/they] did not want to work ...”												

Inflecting Languages

- One morpheme can express several morphological features, e.g. Latin:

lauda (“praise”) + *tur*:

- 3rd person
- singular
- passive voice
- (present tense)

Incorporating (Polysynthetic) Languages

- “Words” are very complex, many features are “incorporated” into the verb, e.g. Yup’ik Inuit:

tuntu	ssur	qatar	ni	ksaite	ngqiggte	uq
reindeer	hunt	FUT	say	NEG	again	3SG:IND

’He had not yet said again that he was going to hunt reindeer.’

Source: Eliza Orr, cited by T. Payne (1997): *Describing morphosyntax: A guide for field linguists*. Cambridge; New York: Cambridge University Press, p. 27-28. ^a

- Are such analyses using a useful notion of “word”?

^a<http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsAPolysyntheticLanguage.htm>

Means of Inflection and Derivation

Affixation:	call+ s ; call+ ed ; call+ ing
Circumfixation:	steigen \Rightarrow stieg \Rightarrow ge · stieg · en
Vowel Changes:	take \Rightarrow took sleep \Rightarrow slep · t weep \Rightarrow wep · t
Vocalization Patterns	kataba — ‘he wrote’ yaktubu — ‘he writes’ kitāb — ‘book’
Reduplication:	paideu · omai \Rightarrow pe · paideu · ka

Concatenation — Assimilation — Fusion

E.g. Japanese:

Concatenation tabe + ta → tabe.ta

Assimilation yari + ta → yat.ta

Fusion kogi + ta → koida

Two-level Morphology (Koskenniemi, 1983)

<http://www.ling.helsinki.fi/koskenni/esslli-2001-karttunen/>

- Two levels of representation: Lexical level and surface level.
- Realization rules have access to both levels.
- Rules operate in parallel, not sequentially.

Two-level Morphology (cont'd)

spy + s → spies

- Insert epenthetic 'e' between ['y' + morpheme boundary] and 's'.
- lexical 'y' → surface 'i' before epenthetic 'e'.

kaNpat → kammat

- 'N' → 'm' before lexical 'p'.
- 'p' → 'm' after surface 'm'.

Y & W's General Approach

- Define probabilistic estimators that evaluate the probability that two word forms are different inflectional forms (e.g. PAST/PRESENT) of the same word.
- Use a combination of these estimators to align the word forms of a language in a probabilistic manner, maximizing the alignment probability under certain constraints (e.g., pidgeon hole principle).
- This alignment can be used as input for a supervised morphological analysis learner.

Alignment by Frequency Similarity

Intuition:

The ratio of word form occurrences, e.g. past tense forms vs. present tense forms is roughly the same across words.

Consequence:

We can use the ratios obtained by counting **regular** word forms in a corpus as estimates of the expected values for irregular word forms.

Alignment by Frequency Similarity

Required:

- Knowledge about regular morphology.
- A large corpus to obtain counts.

Procedure:

- Obtain counts for each pair of related regular forms (e.g. VBD/VB).
- Discretize count frequencies of $\log \frac{\#VBD}{\#VB}$ values to obtain a histogram.
- Fit a curve to the histogram to obtain an estimated probability density function.

Alignment by Frequency Similarity

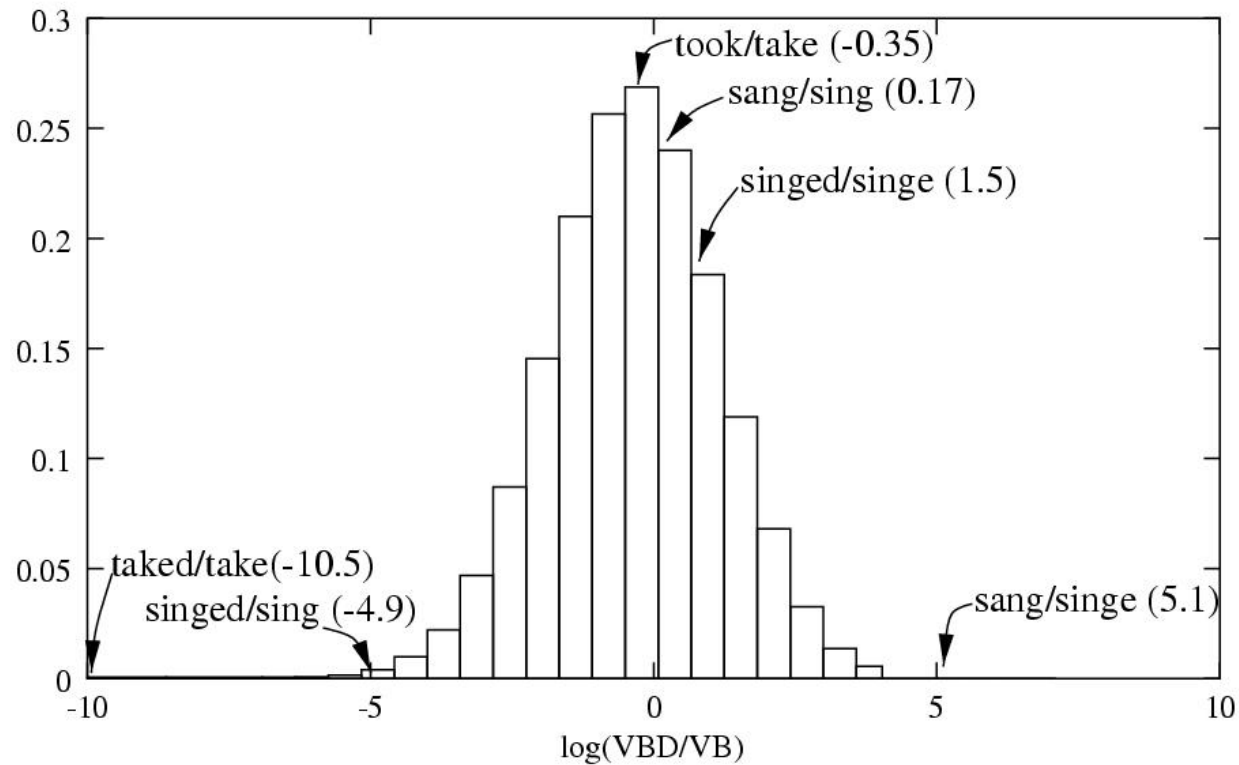


Figure 1: Using the $\log(\frac{VBD}{VB})$ estimator to rank potential VBD-VB pairs in English

from: Yarowsky & Wicentowski (2000)

Alignment by Frequency Similarity

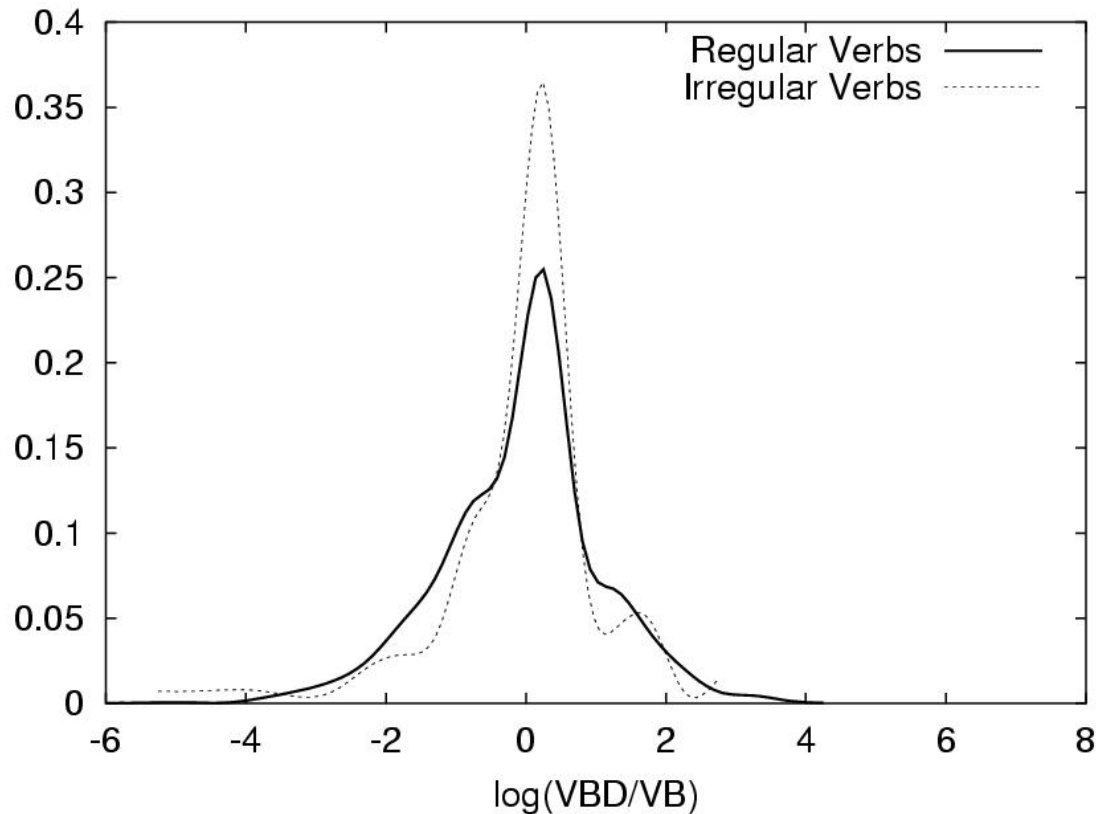


Figure 2: Distributional similarity between regular and irregular forms for VBD/VB

from: Yarowsky & Wicentowski (2000)

Alignment by Context Similarity

- Uses cosine between context vectors of word forms to measure their relatedness.
- Assumes that similarity is highest between different forms of the same word.
- Counting focusses on head nouns of objects and subjects of verbs.
- Uses regular expressions to obtain (very noisy!) counts — no parsing performed.

$$CW_{\text{subj}} (\text{AUX|NEG})^* V_{\text{keyword}} \text{DET? } CW^* CW_{\text{obj}}$$

- Unlike frequency similarity, does not provide information about the exact nature of the relation between the two words.

Regular Expressions for Counts

$CW_{\text{subj}} (AUX|NEG)^* V_{\text{keyword}} DET? CW^* CW_{\text{obj}}$

This **paper** presents an original and successful **algorithm** for ...

Alignment by Weighted Levenshtein Distance

- Levenshtein distance is an edit distance between a source and target word.
- Distance equal to the cost of inserting and deleting characters in source to produce target.
- Typically, cost (weights) of insertion or deletion is 1.
- Here author arbitrarily sets (initial) weights to reflect assumptions about the mutability of v to v, v to c, etc.
- Other suggested initializations: related languages and phonetic similarities.

Cost Matrix

- $\delta_1 = \text{vowel} \leftrightarrow \text{vowel} = 0.5$
- $\delta_2 = \text{vowel cluster} \leftrightarrow \text{vowel cluster} = 0.6$
- $\delta_3 = \text{consonant} \leftrightarrow \text{consonant} = 1.0$
- $\delta_4 = \text{consonant} \leftrightarrow \text{vowel cluster} = 0.98$
- How were these values chosen? and how can the differences between δ_1 / δ_2 and δ_3 / δ_4 make a difference? Does it matter what we initialize with?

Cost Matrix, cont'd

	a	o	ue	m	n	...
a	0	δ_1	δ_2	δ_4	δ_4	...
o	δ_1	0	δ_2	δ_4	δ_4	...
ue	δ_2	δ_2	0	δ_4	δ_4	...
m	δ_4	δ_4	δ_4	0	δ_3	...
n	δ_4	δ_4	δ_4	δ_3	0	...
...

Cost Matrix Updates

- Cost matrix updated iteratively.
- Updates: character to character stem-change probs from current best alignments.
- No final cost matrix given. Number of iterations until convergence not given.

Alignment by Morphol. Transformation Probabilities

- Produces generative probabilistic model.
- Model: $P(\text{inflection} \mid \text{root, suffix, POS})$
- Knowing the stem change is equivalent to knowing the inflection.
- Model: $P(\text{stem change} \mid \text{root, suffix, POS})$

Alignment by Morphol. Transformation Probabilities

- Generate table with the counts of root contexts of various lengths and stem changes.
- E.g. solidify: 'ify', 'fy', 'y'
- Notation: $last_3(solidify) = \text{'ify'}$
- root = $\gamma\alpha$, suffix = $+\sigma$, inflection = $\gamma\beta\sigma$
- Model: $P(\text{stem change} \mid \text{root, suffix, POS})$
- Model: $P(\alpha \rightarrow \beta \mid \gamma\alpha, +\sigma, POS)$

Sample Table

Root Context	Stem Change	Suffix	Count	E.g.
...ify	y→i	+ed	23	solidify → solidified
...ify	ε → ε	+ing	65	solidify → solidifying
...fy	y→i	+ed	143	solidify → solidified
...fy	ε → ε	+ing	98	solidify → solidifying
...y	y→i	+ed	560	solidify → solidified
...y	ε → ε	+ing	435	solidify → solidifying

Iterpolated backoff model

$$\begin{aligned} & P(\alpha \rightarrow \beta | \gamma\alpha, +\sigma, POS) \\ = & \lambda_1 P(\alpha \rightarrow \beta | last_3(r), suf, POS) \\ & + (1 - \lambda_1)\lambda_2 P(\alpha \rightarrow \beta | last_2(r), suf, POS) \\ & + (1 - \lambda_2)\lambda_3 P(\alpha \rightarrow \beta | last_1(r), suf, POS) \\ & + (1 - \lambda_3)\lambda_4 P(\alpha \rightarrow \beta | suf, POS) \\ & + (1 - \lambda_4)P(\alpha \rightarrow \beta) \end{aligned}$$

- Not limited to λ_4 , but how long are ‘very long root contexts’?

Initialization

- No inflection/root pairs exist at initialization.
- Initialize via Levenshtein, with geometric increase in cost from end of root.
- $\text{new cost} = \text{old} * (1 + \text{penalty})$
- $\text{penalty} = \text{param} * (\text{length}(\text{inflection}) - \text{position})$
- Language dependent change in cost: English vs. Spanish

Parameter Re-estimation

- Training examples weighted by their ‘alignment confidence’??
- Must have minimal number of stem change/inflection pairs to keep in model.
- Final model:

$$\begin{aligned} &P(\alpha \rightarrow \beta | root, suffix, POS) \\ &= \lambda_j P_0(\alpha \rightarrow \beta | suffix) \\ &\quad + (1 - \lambda_j) P_j(\alpha \rightarrow \beta | root, suffix, POS) \end{aligned}$$

- Where j is really $1 \rightarrow$ nbr root contexts?
- How are λ updated exactly?
- Same λ as before? Or a backoff model of a backoff model?

Evaluation

Candidate Roots for the English inflection **TOOK** (1st iteration):

Overall Similarity (Iteration 1)				Context Similarity	Frequency Similarity	Levenshtein Similarity	MorphTrans Similarity (1)		MorphTrans Similarity (C)
take	.00162	3.8	1	take .849	take .072	toot .333	toot .002593	take .465578	
turn	.00081	8.7	2	turn .546	tell .028	tool .333	tool .002593	toot .001296	
tell	.00063	15.9	3	tower .332	turn .016	toe .310	tong .000096	tool .001296	
test	.00041	19.6	4	touch .324	talk .014	take .290	tone .000096	tong .000048	
talk	.00051	21.0	5	tip .261	test .001	top .236	...	tone .000048	
tie	.00044	26.7	6	tie .260	teach .001	toil .236	take .000006	tout .000048	

Candidate Roots for the English inflection **SHOOK** (1st iteration):

Overall Similarity (Iteration 1)				Context Similarity	Frequency Similarity	Levenshtein Similarity	MorphTrans Similarity (1)		MorphTrans Similarity (C)
shake	.00149	5.5	1	shake .854	share .073	shoo .500	shoot .002593	shake .465578	
shoot	.00126	9.3	2	shave .323	ship .068	shoot .333	shoo .002593	shoot .001296	
ship	.00104	16.3	3	shape .210	shift .062	shoe .310	shock .000096	shoo .001296	
shatter	.00061	18.9	4	shore .194	shop .060	shake .290	short .000096	shock .000048	
shop	.00094	19.8	5	shower .184	shake .058	shop .236	shout .000095	short .000048	
shut	.00081	20.6	6	shoot .162	shut .052	shout .236	...	shove .000048	
shun	.00039	20.7	7	shock .154	shoot .051	show .236	shake .000003	shore .000048	

Candidate Roots for the Spanish inflection **JUEGAN** (1st iteration):

Overall Similarity (Iteration 1)			Context Similarity	Frequency Similarity	Levenshtein Similarity	MorphTrans Similarity (1)	
jugar	.0024	1	jugar .88	jugar .063	jugar .50	jugar .00129	
juzgar	.0006	2	juntar .38	juzgar .015	juzgar .29	jogar .00129	
jurar	.0002	4	jurar .26	jogar .009	juntar .25	juntar .00004	
jogar	.0000	5	justificar .22	juntar .004	jurar .18	juzgar .00004	

Table 8: Example performance of independent and combined similarity measures

Evaluation

Combination of Similarity Models	# of Iterations	All Words (3888)	Highly Irregular (128)	Simple Concat. (1877)	Non-Concat. (1883)
FS (<i>Frequency Sim</i>)	(Iter 1)	9.8	18.6	8.8	10.1
LS (<i>Levenshtein Sim</i>)	(Iter 1)	31.3	19.6	20.0	34.4
CS (<i>Context Sim</i>)	(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

Table 9: Performance of combined alignment models on 4 classes of past-tense English verbs

from: Yarowsky & Wicentowski (2000)

What I Like About This Paper

- Uses a combination of techniques.
- Doesn't throw out Linguistics.
- Epistemological appeal: How much can we squeeze out of the data?

Concerns

- Is morphological analysis of irregular forms really an unsolved problem?
 - The approach requires some morphological a-priori knowledge anyway, so we can't do without a 'grammar'.
 - Most grammars contain lists and rules for irregular morphological forms (e.g., list them as exceptions).
 - Why not just hack them in and get on with life?
- Separation of training and testing data???
- Are we really achieving anything (beyond insight)?
- Do we really want to achieve anything, beyond insight?

Summary

- Y&W define four estimators that can estimate the probability that two word forms are related forms of the same word (e.g., PRESENT vs PAST).
- They view morphological analysis primarily as an alignment of the word form vocabulary of a large corpus.
- Their method achieves an overall accuracy of 99.2%

“I have demonstrated the effectiveness of this analyzer in Spanish, Portuguese, French, Catalan, Occitan, Italian, Romanian, Latin, English, German, Dutch, Danish, Norwegian, Swedish, Icelandic, Czech, Polish, Russian, Irish, Welsh, Greek, Hindi, Sanskrit, Estonian, Finnish, Turkish, Uzbek, Tamil, Basque, Tagalog, Swahili, and Klingon. In addition, I have done unpublished work in Amharic, Serbo-Croatian, Farsi, Arabic and Malay.”

from Richard Wicentoski's web page