Morpheme segmentation and UR Acquisition with UR Constraints

Max Nelson

University of Massachusetts Amherst

January 6, 2019

The Problem

1. Children must learn to identify word and morpheme boundaries, but must also learn underlying representations and the phonological grammar
2. Phonological cues are used to aid segmentation as early as 8 mos. (Johnson and Jusczyk, 2001), but segmentation errors persist as late as 20 mos. (Babineau and Shi, 2011)
3. Children begin forming lexical representations as early as 6 mos. (Bergelson and Aslin, 2017) and respond to phonological errors by 18 mos. (Swingley and Aslin, 2000)
4. Segmentation must be learned simultaneously with phonological grammar and underlying forms

Segmentation and UR Acquisition

1. Existing UR learners take the set of surface forms as a starting point (Alderete et al., 2005; Merchant and Tesar, 2008; Jarosz, 2015) - implying that segmentation is learned before URs
2. Existing statistical models of segmentation do not make use of lexical representations or phonology beyond phonotactics (Brent and Cartwright, 1996; Goldwater et al., 2009; Daland, 2013; Exceptions include Naradowsky and Goldwater, 2009; Narasimhan et al., 2015; Johnson et al., 2015)
3. Segmentation and URs are learned in parallel and are mutually informing

Segmentation and UR Acquisition

1. Consider adult-like segmentation of novel words in non-novel contexts:

$\sqrt{\text{LOOK}}$	AT	DEF		PL	
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
/lʊk/	/æt/	/ðə/		/z/	
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	*
lʊk	æt	ðə	$\boxed{\text{wʌ}}$	$\boxed{\text{ks}}$	
$\sqrt{\text{LOOK}}$	AT	DEF		PL	
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
/lʊk/	/æt/	/ðə/		/z/	
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	✓
lʊk	æt	ðə	$\boxed{\text{wʌk}}$	$\boxed{\text{s}}$	

UR Constraints

1. Specify the UR for an input, which has no phonological content (Apoussidou, 2007; Pater et al., 2012; Smith, 2015)
2. Candidates are (Input, UR, SR) triplets
3. URs are selected in parallel with phonological optimization, allowing phonological "consequences" of a UR to affect its likelihood
   - Choosing a non-default UR and mapping faithfully is a viable repair strategy

{IND} + ANT	DEP	MAX	HIATUS	IND=/ə/	IND=/ən/
a. ə+ænt → əænt			*W	L	*W
☞ b. ən+ænt → ənænt				*	
c. ə+ænt → ənænt	*W			L	*W
d. ən+ænt → əænt		*W	*W	*	

Current Model

Overview

Goal: Learn phonological alternations, URs (as weighted URCs), and segmentation in parallel

1. URs are stored as URCs which are induced from observed strings
2. Candidates for an input set of MS features are generated from the URCs
3. A Maximum Entropy Grammar (Goldwater and Johnson, 2003) is learned, defining a probability distribution over UR-SR mappings and correspondence relations given an input set of morphosyntactic (MS) features

Current model

UR Constraint induction

1. Given observed string S and corresponding meanings $M_1 \ldots M_n$
2. For every exhaustive segmentation of S that yields n nonempty substrings $s_1 \ldots s_n$:
   - For c in the set of UR constraints of the form $M_{1 \ldots n}{=}/s_{1 \ldots n}/$:
     - If c not in $\textsc{Con}$, add c to $\textsc{Con}$ with weight w
3. Example, {M1,M2}→[abc]:

Segmentation	Constraints added
a.bc	M1=/a/, M2=/a/, M1=/bc/, M2=/bc/
ab.c	M1=/ab/, M2=/ab/, M1=/c/, M2=/c/

Current model

Assumptions

1. The learner is provided with the number of morphosyntactic features in a string
   - Segmentation is simplified, not uncommon in morphology induction (Naradowsky and Goldwater 2009; Narasimhan et al. 2015)
   - IO correspondence relations are not provided, removing an assumption of previous UR learners
2. For every morpheme there must be at least one surface form that is a faithful mapping from the underlying form
3. Every morpheme in the input must have a correspondent in the output
4. Every segment in the output must be associated with some morpheme in the input
5. The set of segments corresponding to a single morpheme must be contiguous

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 ... M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times ... \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c
a. /**a$_1$.bc$_2$**/		-1		-1

Current model
Candidate Generation

1. UR_n is the set of all URs specified by URCs in $\textsc{Con}$ for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c
a. /**a$_1$.bc$_2$**/		-1		-1
b. /a$_1$.c$_2$/		-1	-1	

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c
a. /**a$_1$.bc$_2$**/		-1		-1
b. /a$_1$.c$_2$/		-1	-1	
c. /ab$_1$.bc$_2$/	-1			-1

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

$\{M1_1, M2_2\}$	$\{M1\}$=a	$\{M1\}$=ab	$\{M2\}$=bc	$\{M2\}$=c
a. /**a**$_1$**.bc**$_2$/		-1		-1
b. /a$_1$.c$_2$/		-1	-1	
c. /ab$_1$.bc$_2$/	-1			-1
d. /**ab**$_1$**.c**$_2$/	-1		-1	

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in Con for M_n
2. For an input $M_1 ... M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times ... \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c
a. /**a$_1$.bc$_2$**/		-1		-1
b. /a$_1$.c$_2$/		-1	-1	
c. /ab$_1$.bc$_2$/	-1			-1
d. /**ab$_1$.c$_2$**/	-1		-1	

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

$\{M1_1, M2_2\}$	$\{M1\}$=a	$\{M1\}$=ab	$\{M2\}$=bc	$\{M2\}$=c
a. /a_1.bc_2/→[abc]		-1		-1
b. /a_1.c_2/→[ac]		-1	-1	
c. /ab_1.bc_2/→[abbc]	-1			-1
d. /ab_1.c_2/→[abc]	-1		-1	

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in Con for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c	Max(a)
a. /a$_1$.bc$_2$/→[abc]		-1		-1	
b. /a$_1$.c$_2$/→[ac]		-1	-1		
c. /ab$_1$.bc$_2$/→[abbc]	-1			-1	
d. /ab$_1$.c$_2$/→[abc]	-1		-1		

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c	MAX(A)
a. /a$_1$.bc$_2$/→[abc]		-1		-1	
b. /a$_1$.c$_2$/→[ac]		-1	-1		
c. /ab$_1$.bc$_2$/→[abbc]	-1			-1	
d. /ab$_1$.c$_2$/→[abc]	-1		-1		
e. /a$_1$.bc$_2$/→[bc]		-1		-1	-1
f. /a$_1$.c$_2$/→[c]		-1	-1		-1
g. /ab$_1$.bc$_2$/→[bbc]	-1			-1	-1
h. /ab$_1$.c$_2$/→[bc]	-1		-1		-1

Learning algorithm

1. Online, error driven, stochastic gradient descent
2. Minimizing negative log likelihood of data, no regularization
3. In standard MaxEnt learning:

$$\delta w_i \propto \overbrace{c_i(y)}^{\textit{Observed}} - \overbrace{\sum_{x \in \Omega_M} c_i(x)p(x)}^{\textit{Expected}}$$

4. However we don't know $c_i(y)$, because the observed mapping {M1,M2}→[abc] does not provide direct information about the UR or segmentation

$\{M1_1, M2_2\}$	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c	DEP(C)
a. /a_1.bc_2/→[abc]		-1		-1	
b. /ab_1.c_2/→[abc]	-1		-1		
c. /a_1.b_2/→[abc]		-1		-1	-1

Expectation Maximization

Probabilistic URs and segmentation

1. Expectation maximization, general algorithm for MLE with incomplete data (Dempster et al., 1977)
2. History of application to phonological learning with structural ambiguity (Tesar and Smolensky, 1998; Jarosz, 2006; Pater et. al., 2012)
3. The E step assigns a probabilistic structure to the observed form, the M step updates as normal, maximizing the probability of the structure assigned in E
   - E:

$$\hat{c}_i(y) = \sum_{z \in Z_y} c_i(z) \frac{p(z)}{\sum_{z \in Z_y} p(z)}$$

   - M:

$$\delta w_i = \hat{c}_i(y) - \sum_{x \in \Omega_M} c_i(x) p(x)$$

Test case: English Plural

English Phrase	Input String	Input Morphemes
a dog	ədɔg	IND, DOG
the dog	ðədɔg	DEF, DOG
the dogs	ðədɔgz	DEF, DOG, PL
a cat	əkæt	IND, CAT
the cat	ðəkæt	DEF, CAT
the cats	ðəkæts	DEF, CAT, PL
a pie	əpai	IND, PIE
the pie	ðəpai	DEF, PIE
the pies	ðəpaiz	DEF, PIE, PL

English Plural

Possible solutions

1. The plural morpheme is underlyingly /z/ and devoices following voiceless
   - PL=/z/ and AGREE are high
   - ID(VOI) and other URCs for PL are low
2. The plural morpheme underlyingly alternates between /z/ and /s/ to map faithfully without violating AGREE
   - AGREE and ID(VOI) are high
   - PL=/z/ and PL=/s/ are low with PL=/z/ above PL=/s/

Test case: English Plural

1. 2,000 iterations with a learning rate of 0.1 and all weights initialized at 1.0
2. In all phrases the probability of correct segmentation candidates is above 0.98

Constraint	Weight
PL=/z/	10.61
IND=/ə/	9.15
AGREE	8.96
DOG=/dɔg/	8.72
CAT=/kæt/	8.52
DEF=/ðə/	7.92
PIE=/pai/	7.67
ID(VOI)	3.60
PL=/s/	1.12
...	< 0.065

Test case: English Plural

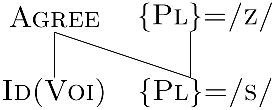
1. 2,000 iterations with a learning rate of 0.1 and all weights initialized at 1.0
2. In all phrases the probability of correct segmentation candidates is above 0.98

Constraint	Weight
IND=/ək/	0.06
DEF=/ðəp/	0.009
DOG=/g/	$9.40E{-}5$
PL=/gz/	$1.76E{-}5$
CAT=/kæ/	$1.56E{-}5$
PIE=/p/	$3.13E{-}6$
PIE=/i/	$2.17E{-}6$
IND=/ədɔ/	$1.31E{-}6$
...	$< 1.31E{-}6$

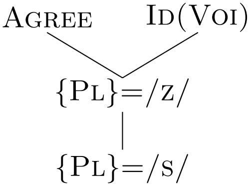
Why assimilation and not allomorphy?

1. Recall that with URCs we can choose an alternative UR rather than violate FAITH
2. In 97 of 100 runs assimilation is learned
3. Weighting arguments for assimilation are two-tiered, for allomorphy are three-tiered
4. Randomly initialized weights between 0 and 5 satisfy assimilation 14.68% of the time, allomorphy 3.86%

Assimilation:



Allomorphy:



Segmenting novel words

1. The final grammar can be used to segment novel words in familiar contexts
2. Below are segmentation candidates for {WUG, PL}→[wʌgz] and {WUK, PL}→[wʌks]

UR	SR	Probability	UR	SR	Probability
/wʌg/+/z/	wug.z	0.9853	/wʌk/+/z/	wʌk.s	0.9413
/wʌg/+/s/	wug.z	0.0020	/wʌk/+/s/	wʌk.s	0.0198
/wʌ/+/gz/	wu.gz	0.0049	/wʌ/+/kz/	wʌ.ks	0.0015
/wʌ/+/gs/	wu.gz	0.0015	/wʌ/+/ks/	wʌ.ks	0.0046
/w/+/ʌgz/	w.ugz	0.0049	/w/+/ʌkz/	w.ʌks	0.0015
/w/+/ʌgs/	w.ugz	0.0015	/w/+/ʌks/	w.ʌks	0.0046

Conclusions

1. Morpheme identity is a type of hidden structure
2. A joint model is able to learn URs, segmentation, and alternations. The final grammar is able to segment novel words in non-novel environments
3. An explicit mechanism to learn segmentation may not be necessary given learning of URs and IO correspondence relations

Thank you

Particular thanks to Katherine Blake, Gaja Jarosz, Andrew Lamont, Joe Pater, Brandon Prickett, UMass Sound Workshop, and everyone at NECPHON 2018

Weighting arguments for assimilation and allomorphy

Assimilation:	Allomorphy
AGREE > ID(VOI)	PL=/z/ > PL=/s/
PL=/z/ > PL=/s/	ID(VOI) + PL=/s/ > PL=z
PL=/z/ > ID(VOI) + PL=/s/	AGREE + PL=/s/ > PL=z
PL=/z/ + ID(VOI) > PL=/s/	PL=/z/ + ID(VOI) > PL=/s/
	PL=/z/ + AGREE > PL=/s/