Learning Morphological Productivity as Meaning-Form Mappings

Sarah Payne
University of Pennsylvania
paynesa@sas.upenn.edu

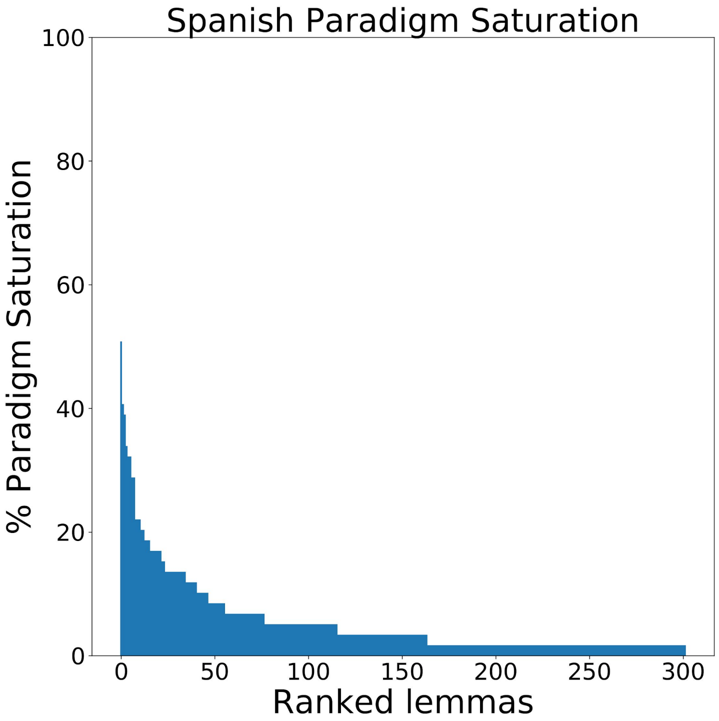
Jordan Kodner
Stony Brook University
jordan.kodner@stonybrook.edu

Charles Yang
University of Pennsylvania
charles.yang@ling.upenn.edu

The Problem

- Children learn the entirety of verbal morphology from very sparse input
- They have no explicit information as to whether their language is agglutinative or fusional
- This is a mapping problem:

  Semantic features ➜ morphological features



Our Approach

- We collect child-directed verb forms from CHILDES for English and Spanish
- We annotate these using UniMorph tags
    - UniMorph provides person, number, tense, etc; we consider this an approximation of the child's semantic knowledge
- We apply the Tolerance Principle recursively on the data to pick out larger and smaller patterns (more on this later)

Outline

- Data + Spanish Basics
- Tolerance Principle + Model
- Results

Data

Data: Spanish and English

- **Spanish: 989 inflected forms, 302 lemmas**
  - Sampled from FernAguado corpus by frequency
  - Example:
    - tener   V;IND;PRS;2;SG   tienes
- **English: 3,953 inflected forms, 1,285 lemmas**
  - Sampled from Manchester, Wells, and Belfast corpora by frequency
  - Example:
    - bake   V;V.PTCP;PRS   baking
- Frequency is correlated with irregularity in English, but not Spanish (Fratini et al. 2014)

Spanish Basics

- 3 main classes of verbs: *-ar*, *-ir* and *-er* (defined by infinitive form)
    - *-ar* is largest class (62% of our data vs. 24% *-er* and 14% *-ir*)
    - Mappings often correspond to its behaviour
- Tense and person+number are often indicated separately in an agglutinative fashion
    - E.g., ***-ria*** = **COND**, ***-ra*** = **FUT**, ***-ba*** = **IPFV** (-ar verbs), and ***-s*** = **[2; SG]**
    - So ***-rias*** = **[COND; 2; SG]**, ***-ras*** = **[FUT; 2; SG]**, ***-bas*** = **[IPFV; 2; SG]**

Model

The Tolerance Principle (Yang, 2016)

$$e \leq \theta_N = \frac{N}{\ln N}$$

N = the number of words to which are eligible to take a rule

e = the number of those words to which the rule does not apply

Example: 100 past-tense English verbs; 20 don't take ***-ed***. 100/ln100 = 21.7. 20 < 21.7 so **PST** ➔ ***-ed*** passes the Tolerance Principle

Model Overview

- GCD application of the Tolerance Principle
- Recursive application of the Tolerance Principle
- Multi-pass application of the Tolerance Principle

Method: GCD approach

- **Possible suffixes** = substrings of length ≤ 5 at the end of an inflected form
    - e.g., possible suffixes of *ama**remos*** are ***-remos**, **-emos**, **-mos**, **-os**, **-s***
- **Possible features realized by each suffix** = all possible subsets of the provided feature set
    - e.g. possible features for **[IND; PRS; 3; SG]** could be **[IND], [IND; PRS], [IND; PRS; 3], [IND; PRS; 3; SG], [PRS], [PRS;3] ...**
- Use a **GCD approach:** find smallest feature-set that maps to a suffix
    - Do a pass of the TP from feature-sets to suffixes
    - For each suffix that was mapped to, find the intersection of all features that mapped to it
    - Keep adding features by frequency until a mapping from the features to suffix passes

Example: GCD approach using the Tolerance Principle

In Spanish, ***-mos*** = **[1;PL]**, which we obtain as follows:

1. Do a pass of the TP from feature-sets to suffixes
    - This yields mappings such as **[1; PL]** = ***-mos***, **[1; PL; FUT]** = ***-ramos***, **[1; COND]** = ***-riamos***
    - Some of these (e.g. **[1;COND]**) are underspecified, others are overspecified
    - We cannot learn agglutinativity from these mappings alone

Example: GCD approach using the Tolerance Principle

2. For each suffix that was mapped to, find the intersection of all features that mapped to it
    - For ***-mos,*** say this suffix was mapped to by **[1;PL], [IND; PRS; 1; PL],** and **[POS; IMP; 1]**
    - The intersection of these gives **[1]** = ***-mos***, which won't pass the TP

Example: GCD approach using the Tolerance Principle

3. Keep adding features by frequency until a mapping from the features to suffix passes
   - The second-most frequent feature is **PL**, and **[1; PL]** = ***-mos*** passes

Method: Recursive Application of the TP

- We learn the broadest mappings first
  - e.g. in Spanish, **[3; SG]** = **“”**
- Then we recurse on the exceptions to these broad mappings to learn narrower mappings
  - e.g. in Spanish, **[3; SG]** = **“”** except **[3; SG; PFV]** = ***-o***
  - We learn the latter mapping by recursively applying the TP to the verbs that fail to be correctly inflected by **[3; SG]** = **“”**
- We memorize verbs that remain exceptions after recursion
  - In Spanish, we learn narrow mappings such as **[3; SG; PFV]** = ***-o*** and stem conditioned endings such as the imperfective
  - In both cases, we can predict the rule we use based on properties of the lemma or features
  - However, we can’t do the same for *ser*, so we memorize its inflected forms

Method: Multi-Pass Application of the TP

- In agglutinative languages, more frequent features are realized closer to the end of the inflected form
  - e.g. in Spanish, person & number are always realized at the end and are most common
- We consider feature categories (person, number, mood, tense, aspect) in order of decreasing frequency
- At each pass, we constrain GCD mappings to the given feature category/categories and recurse on these before moving to the next one
  - In Spanish, we learn person-number endings and their productive exceptions first.
  - This includes **[3; SG]** = **“”** and **[3; SG; PFV]** = ***-o***, which is learned via recursion at this pass
- We remove the suffixes we’ve learned at a given pass from the inflected forms before moving on to the next pass
  - After removing Spanish person-number endings, we learn mappings such as **[COND]** = ***-ria***

Model Overview

At each pass, constrained by feature categories:

- Do a GCD pass of the Tolerance Principle
- Recurse on exceptions
- Memorize anything left

Results

Results: English

Broad Mappings				
Features	**Defau.**	**Alternations**	**Ct.**	**Ex.**
First Pass				
PRS	∅		2573	*walk*
Second Pass				
3	∅		1717	*walk*
2	∅		571	*walk*
1	∅		554	*walk*
Third Pass				
PL	∅		1454	*walk*
SG	∅		1422	*walk*
Fourth Pass				
NFIN	∅		22	*walk*

Narrow Mappings				
Features	**Defau.**	**Alternations**	**Ct.**	**Ex.**
First Pass				
PTCP, PRS	ing	e → ing	643	*pleasing*
3 SG PRS	s		372	*walks*
Second Pass				
3 PL PST	ed	y→ied,e→ed	367	*pleased*
3 SG PST	ed	y→ied	139	*tried*
2 SG PST	ed	y→ied,e→ed	203	*walked*
1 SG PST	ed	y→ied,d→t	136	*built*
1 PL PST	ed	y→ied	67	*cried*

Results: Spanish

Broad Mappings				
Features	**Default**	**Alterns.**	**Ct.**	**Ex.**
First Pass				
3 SG	∅		227	*ama*
3 PL	n		103	*aman*
1 PL	mos		51	*amamos*
2 PL	is		10	*amais*
PRS 1 SG	o		163	*amo*
PRS 2 SG	s		129	*amas*
Second Pass				
IND	∅		651	*ama*
IMP	∅		127	*ama*
NFIN	r		146	*amar*
COND	ria		16	*amaria*
Third Pass				
PRS	∅		492	*ama*
FUT	ra		20	*amara*
Fourth Pass				
IPFV	ia	a→aba	65	*amaba*

Narrow Mappings				
Features	**Default**	**Alterns.**	**Ct.**	**Ex.**
First Pass				
SBJV PRS 3 SG	e	i → a	13	*ame*
POS IMP 3 SG	e	i → a	14	*ame*
IND PST 3 SG PFV	o		72	*amo*
SBJV PRS 3 PL	an		2	*coman*
POS IMP 3 PL	an		2	*coman*
IND PST 3 PL PFV	ron		23	*amaron*
POS IMP 1 PL	emos		3	*amemos*
SBJV PRS 1 PL	emos		3	*amemos*
POS IMP 2 PL	d		2	*amad*
SBJV PRS 1 SG	e	i → a	14	*ame*
IND PST 1 SG PFV	e	i → i	18	*ame*
COND 2 SG	rias		2	*amarias*
SBJV PRS 2 SG	es	i → as	33	*ames*
IND FUT 2 SG	ras		3	*amaras*
IND PST 2 SG IPFV	ias		9	*comias*
IND PST 2 SG PFV	ste		10	*amaste*
Second Pass				
IND FUT 1 PL	re		2	*amaremos*

Discussion + Future Work

- Segmentation and generation
  - Our model may be extended to be competitive on computational linguistics and NLP morphological tasks
- Developmental plausibility
  - Our model learns rules in a similar order to children
  - Does it exhibit U-shaped development?
- Non-verbal morphology
  - Derivational or German nouns

Thank you!!

We’d also like to thank Bob Berwick and his lab, Spencer Caplan, Kyle Gorman, Mitch Marcus, Hongzhi Xu, and the anonymous SCiL reviewers for their feedback.