Multilingual Learning as Parameter Co-occurrence Clustering

Max Bane[1]
bane@uchicago.edu
CAS Workshop on Language and Cognition
University of Chicago
October 26, 2007

1   Introduction

This talk is about "multilingualism" in the broadest sense:

- The knowledge and use of many distinct linguistic systems by individual speakers (listeners).

Multilingualism under this definition is pervasive. Virtually all human beings can distinguish and use multiple linguistic systems — some quite similar to each other, others very different — at various points on the language–dialect–register continuum.
   Some examples:

- Native bilingualism and code-switching.

- "Multi-dialectism". For instance, Clopper (2004) provides an extensive study of American English speakers' abilities to distinguish and categorize multiple dialects, as well as to employ multiple dialects natively, and even imitate them non-natively.

- Biber (1995) gives a detailed cross-linguistic survey of register variation. Language users make use of systematically different phonological, morphological, and syntactic forms — in effect, distinct linguistic systems — depending on social and conversational context.

On this view, what does one know when one knows, say, English? A collection of mostly overlapping but different systems of communication (registers, dialects, others' idiolects), perhaps acquired to different degrees, together with when to use them. A bilingual English-French speaker knows an even larger collection of this sort.
   ⇒ How are people able to acquire, and distinguish among, the member systems of these complex collections?
   I will present a strategy (class of algorithms) for modeling how this might be accomplished within a parameterized linguistic theory. Though the approach outlined here is applicable at any level of linguistic analysis (phonology, morphology, syntax, . . .) and in any theory that has a notion of parameters (with some caveats), for purposes of concreteness and testing/evelution, we'll be working within an optimality theoretic (OT; Prince & Smolensky, 1993) framework for syllable shape phonotactics.
   Outline of the talk:

2 Background: the language learning problem in general, as parameter estimation.

3 The multilingual learning problem in general, as learning from a mixture of languages.

4 The basic strategy: clustering on parameter co-occurrence.

5 Specific algorithms and their evaluation.

6 A preview of what the algorithms imply for population dynamics.

7 Summary and future work.

---

[1]The present author has collaborated extensively with Jason Riggle, James Kirby, and John Sylak on the research presented here.

2   Learning as Parameter Estimation

2.1   Language as a mapping

Linguistic systems are often formalized as mappings from some set of inputs ("underlying representations") to a set of possible outputs ("surface representatins"):

$$L : I \to O$$

For example:

(1)   Syllabification

	Input		*Output*
	CVCCV	$\mapsto$	.CV.CCV.

(2)   Morphology

	Input		*Output*
	PastPart(buy)	$\mapsto$	bought (Most varieties of English)
	PastPart(buy)	$\mapsto$	boughten (Some varieties of English)
	PastPartMasc(acheter)	$\mapsto$	acheté (Most varieties of French)

(3)   Syntax/Semantics

	Input		*Output*
	$\exists t(t < n \land \text{KISS}(\text{John})(\text{Mary})(t))$	$\mapsto$	Mary kissed John. (Most varieties of English)

A simple approach with some advantages:

- Both production and recognition can be approached in terms of the same formal objects (just reverse the mapping).

- Readily extensible. One might model variation by mapping inputs to multiple possible outputs, or by mapping (i, o)-pairs to probabilities.

2.2   Learning from examples of the mapping

Within such a formalization, the problem faced by the learner of a language can be stated as:

- Given some finite sample $S = \{(i_1, o_1), \dots, (i_n, o_n)\}$ of example input-output pairings, what is the mapping L that generated them?

Whatever hypothesis the learner arrives at for L, it should be usable to predict what outputs correspond to inputs that were not seen in the sample (i.e., the learner *generalizes* from S).

Supervised Learning	The learner's sample S contains (input, output)-pairs.
Unsupervised Learning	S contains only outputs. The learner must hypothesize the underlying inputs in addition to the mapping that generates the observed outputs from them.

The learning problem has been studied extensively under both scenarios, though there is some controversy as to what exactly it means for the learner to "observe" inputs in the supervised case. Nonetheless, the present work will assume a supervised setting (following much recent linguistic work, e.g., Tesar & Smolensky 2000, Boersma & Hayes 2001, Riggle 2004).

Thus the learner is represented as some algorithm A that is fed a sample S of (i, o)-pairs, and returns a hypothesized mapping (language) H. If that H happens to be identical to the mapping L from which S was drawn, then we say that A has "learned" L.

2.3   Parameterizing the mapping

For most human linguistic systems, the mapping L, if written out as a list of the (input, output)-pairs that it defines, would be infinite in length (or at least very, very long). For this reason, theories of language

usually work in terms of finite, hopefully concise, sets of *parameters*, different values of which describe or define different mappings.

Any given set of parameter values, g, is called a grammar, and the function (or set of principles, or whatever), $\mathcal{G}$, that determines what mapping, $L = \mathcal{G}(g)$, a given grammar defines is sometimes referred to as "Universal Grammar".

Thus we can restate the task of the supervised learner:

- Given a finite sample $S = \{(i_1, o_1), \ldots, (i_n, o_n)\}$ of input-output pairs, what parameter settings g might define the mapping $L = \mathcal{G}(g)$ that generated them?

This is essentially the same problem as what statisticians call "parameter estimation".

2.4   Optimality Theory and its parameters

OT in three (and a half) bullet points:

- There is a set CON of constraints on possible (i, o) mappings. A given (i, o)-pair might violate any number of these constraints any number of times.

- A grammar g is any total ordering on (i.e., ranking of the constraints in) CON.

- The output assigned for a given input i in the language $\mathcal{G}(g)$ is that output o_H (out of all possible outputs) such that (i, o_H) incurs the fewest violations of the highest ranked constraints.

  - More precisely, we say that (i, o_1) is more "harmonic" than (i, o_2) iff (i, o_1) incurs fewer violations than (i, o_2) of the highest ranked constraint on which they differ. o_H is then that output for which (i, o_H) is pairwise the most harmonic of all possible (input, output)-pairs.

Example: Constraints *x (penalize each occurrence of "x" in the output), *z (penalize each occurrence of "z" in the output), *(x→y) (penalize each occurrence of "y" in the output where there is an "x" in the input). For the grammar (ranking) $g = $ *x $\gg$ *(x→y) $\gg$ *z, we can calculate the output that corresponds to a hypothetical input "xyz" in $\mathcal{G}(g)$ by constructing a tableau:

xyz	*x	*(x→y)	*z
xyz	*		*
xxz	**		
→ zyz			**
yyz		*	*
. . .			

The parameter settings (grammar) that an OT learner infers from a sample of a language are simply a specification of each constraint's place in the total ordering (e.g., a vector of dimensionality $|\text{CON}|$ where the value of the ith component is an integer $j \in \{1, \ldots, |\text{CON}|\}$ specifying that constraint i has rank j in the total ordering). Note that there may be a notion of inconsistent or impossible parameter settings — for example, assigning multiple constraints the same rank.

3   Learning from Mixtures of Languages

Section 2 gives a rough outline of how the process of language learning and questions of learnability are typically formalized and tackled: one target language, of which the learner observes positive examples, applying some algorithm to infer a hypothesis grammar for the target language.

The multilingual learning problem requires only a simple modification to this framework but is significantly more difficult:

- Given a finite sample $S = \{(i_1, o_1), \ldots, (i_n, o_n)\}$ of input-output pairs drawn from *multiple languages*, what *set of grammars* might define those languages according to $\mathcal{G}$?

A major source of difficulty in this problem is illustrated by the hypothetical case in (4).

(4)   a. L_1: /VC/ $\mapsto$ [CVC]
      b. L_2: /VC/ $\mapsto$ [V]

Suppose one language represented in the sample, L_1, epenthesizes (inserts) onsets onto syllables lacking them (4a), while another, L_2, deletes syllable codas (4b). If the learner is to acquire both L_1 and L_2 as distinct languages, then his or her challenge is to avoid generalizing to a grammar of L_3, which does both:

$$L_3 : \text{/VC/} \mapsto \text{[CV]}$$

Existing models of learning variation within a single language (e.g., Boersma & Hayes 2001), if adapted to the present situation where the learning sample is a union of observations from several languages, will do just this. Some other strategy is needed.

Two competing pressures:

- The need to distinguish or separate the languages represented in the sample.

- The need to accomodate the possibility that the target languages might be highly similar, and overlap significantly.

It might be that human learners rely on all sorts of extralinguistic information to help them in this task. This possibility can ultimately be fit into this framework (a simple but effective extension of this sort will be explored in (5.3)), but we'll start from the simplest statement of the problem, where the learner must distinguish between individual languages using only the information contained in a sample of their union.

4   A Strategy: Tracking Parameter Co-occurrence

The strategy I will propose here is related to Biber's "multi-dimensional" analyses of register variation, and some older ideas:

> "[W]hen analyses are based on the co-occurrence and alternation patterns within a group of linguistic features, important differences across registers are revealed. ... Ervin-Tripp (1972) and Hymes (1974) identify 'speech styles' as varieties that are defined by a shared set of co-occurring linguistic features. Halliday (1988:162) defines a register as '*a cluster of associated features having a greather-than-random ... tendency to co-occur*'." Biber (1995, pg. 30), emphasis mine.

This insight can form the basis of a multilingual learning algorithm in the framework described in sections 2 and 3 if we adopt the following assumption:

(5)   Given a single (input, output)-pair, it is possible to determine (efficiently) which parameter settings are consistent with that pair — i.e., which grammars define a language containing that pair.

This requirement can be satisfied within OT if certain basic restrictions are placed on the contents of the constraint set. Statements about which grammars (constraint orderings) are consistent with a given observation (i.e., what properties the grammar must have) can then be formulated as conjunctions of Prince's (2002) Elementary Ranking Conditions (ERCs).

(Briefly: an ERC expresses a disjunction of partial orderings, e.g., "constraints A and B outrank constraints C or D". Thus the conjunction of ERCs implied by an observation is a specification of the what pairwise orderings a constraint ranking must contain in order to be consistent with that observation. A large enough conjunction of ERCs might specify a unique total ordering.)

With this assumption met, the following strategy for multilingual learning becomes possible:

1. Begin with an empty, unweighted, undirected "co-occurrence graph".

2. For each observation (i, o) in the sample:

   (a) Construct a list of statements about which properties a grammar would have to possess in order to be consistent with seeing that observation (in OT, a list/conjunction of ERCs).

(b) For each statement in that list, add a node to the co-occurrence graph, and add an edge between between each pair of those nodes.

After doing this for the entire sample of observations, the co-occurrence graph reflects which grammatical properties were seen to be consistent with the sample, and the edges in the graph indicate which grammatical properties were seen to be *simultaneously required for a single observation*. See Figure 1.

Hypothesis: Intuitively, the "dense" or highly connected, mutually consistent, regions of the graph tend to correspond to the grammars of the individual languages from whose mixture the sample was drawn.

3. Apply some heuristic to identify the dense regions, or clusters, of the co-occurence graph, and adopt hypothesis grammars that are consistent with the grammatical properties specified by those clusters.

Essentially, steps 1 and 2 of the strategy serve to reduce the multilingual learning problem to one that is heavily studied in the fields of computational learning theory and data mining: clustering.

- With the added requirements that the clusters might overlap, and should specify a mutually non-contradictory set of grammatical properties.

Depending on one's clustering heuristic, there is often no need to presuppose the number of languages represented by the sample.
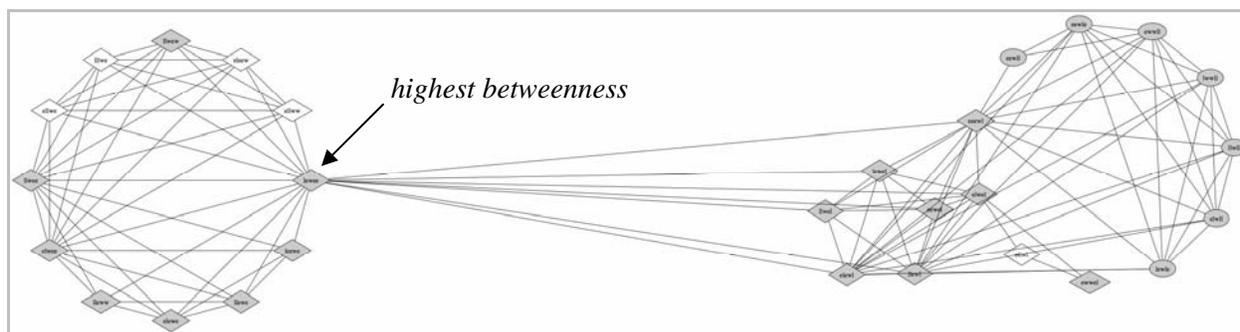

Figure 1: A co-occurrence graph constructed from a sample of the language mixture of two 5-constraint Finite State OT grammars. Note that there are at least two dense regions. The learning strategy would be to adopt these regions of grammatical properties as defining the learner's hypothesis about the grammars that produced the mixed sample.

5   Algorithms and Results

5.1   Clustering on betweenness centrality

Implementing the above strategy algorithmically requires choosing a heuristic for Step 3. There are a variety of possibilities, some quite sophisticated and resulting from long lines of research in a variety of fields, but for an initial assessment of the strategy, I will adopt a fairly simplistic one, based on the graph theoretic notion of "centrality".

The "betweenness centrality" of a node in a graph (Freeman, 1977):

- The betweenness centrality of a node n in a graph G is the proportion of shortest paths between pairs of nodes in G that pass through n.

Intuitively speaking, betweenness centrality offers a means of identifying dense regions in a graph by locating the nodes which lie on their edges — the nodes *between* dense areas. Figure 1 indicates the "betweenest" node in an example co-occurrence graph.

A simple algorithm for Step 3 of the strategy follows:

(6) **Betweenest node deletion heuristic**

   1. Begin with an empty set H of hypothesis grammars.

   2. For each connected component C of the co-occurrence graph G:

      (a) If C represents a set of mutually consistent grammatical properties, construct a hypothesis grammar consistent with C and add it to H. Remove C from G.

      (b) Otherwise, find the node in G with the greatest betweenness centrality, and remove it from G. Return to Step 2.

This is a fine starting point, but has two drawbacks:

- By deleting nodes, we lose much of the information contained in the co-occurrence graph.

- This strategy is unable to yield hypothesis languages that overlap each other, even if the target languages do.

A way of addressing this is to "split" the betweenest nodes apart, distributing copies of them to each cluster that they lie between. This effectively allows overlapping hypotheses, and doesn't lose any information.

(7) **Betweenest node splitting heuristic**

   1. Begin with an empty set H of hypothesis grammars.

   2. For each connected component C of the co-occurrence graph G:

      (a) If C represents a set of mutually consistent grammatical properties, construct a hypothesis grammar consistent with C and add it to H. Remove C from G.

      (b) Otherwise, find the node v in G with the greatest betweenness centrality, and tentatively remove it from G to determine what new connected components $k_1, \ldots, k_n$ appear in G as a result of its loss. Add copies of v back to each of $k_1, \ldots, k_n$, reconnecting those copies to which ever nodes in $k_1, \ldots, k_n$ that v shared an edge with before removal. Return to Step 2.

5.2   Testing the node splitting algorithm

- The test setup:

   - Finite State OT grammars of syllable shape phonotactics, as described by Riggle (2004) — 10 constraints over the alphabet $\{C, V\}$. See (1).

   - A learner applying algorithm (7) to learn the underlying grammars from the mixed data of 1–5 languages.

   - Target language grammars randomly generated each trial.

   - Usually 100 trials; examine average performance.

- Evaluating the results:

   - How best to do so would merit an entire talk.

   - Count the learner's "overgeneralization" errors of the sort in (4).

   - A similarity metric between individual hypotheses and target grammars: for what proportion of a given input lexicon would they be expected to agree on the outputs? (If hypotheses are vague, how often would they be *expected* to agree if the vague bits were resolved uniformly at random?) "Expected agreement."

   - A distance metric between *sets* of hypothesized grammars and the *set* of target grammars. "Expected agreement displacement."

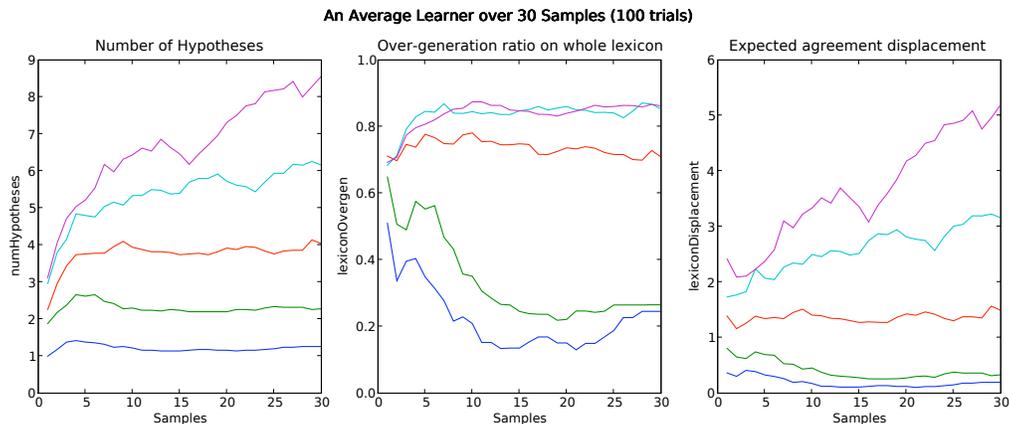- Some preliminary results are charted in Figure 2.

An Average Learner over 30 Samples (100 trials)

Figure 2: The trajectories of an average learner's performance using algorithm (7), as the sample size increases.

5.3 Variations: speaker features and hypothesis merging

Algorithm (7) consistently tends to overestimate the number of languages represented in the multilingual sample.

- Possible solutions:
  - Incorporating extralinguistic information ("speaker features"): the co-occurrence graph includes nodes for the individual "speakers" generating the sample, and connects grammatical properties to the speakers they were seen to come from. Some preliminary results are shown in Figure 3.
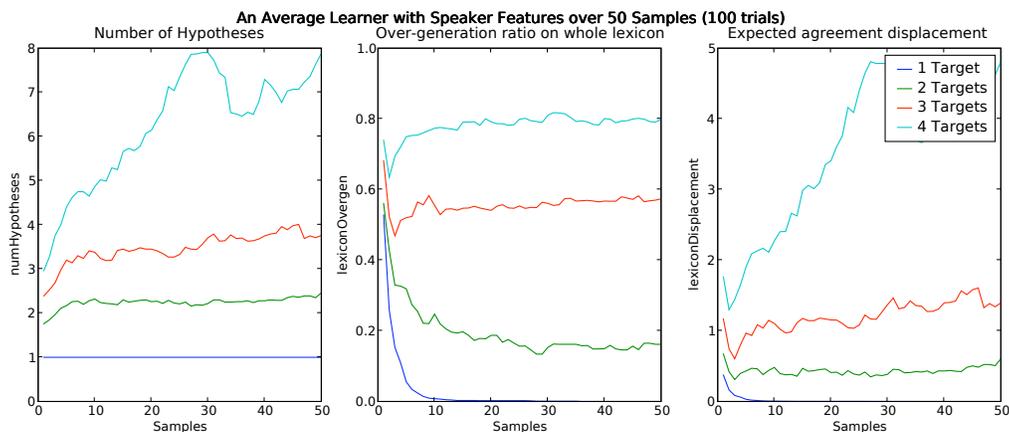 - Hypothesis merging: Add an additional phase of consolidating hyothesized grammars where possible.

An Average Learner with Speaker Features over 50 Samples (100 trials)

Figure 3: The trajectories of an average learner's performance using algorithm (7) together with speaker feature tracking, as the sample size increases.

5.4 Learner population dynamics

- How does the learning algorithm predict that a population of speakers would evolve over time?

- A preliminary investigation: the "telephone" game. The output hypotheses of one learner becomes the source of the input sample for the next learner, iteratively for some number of generations.
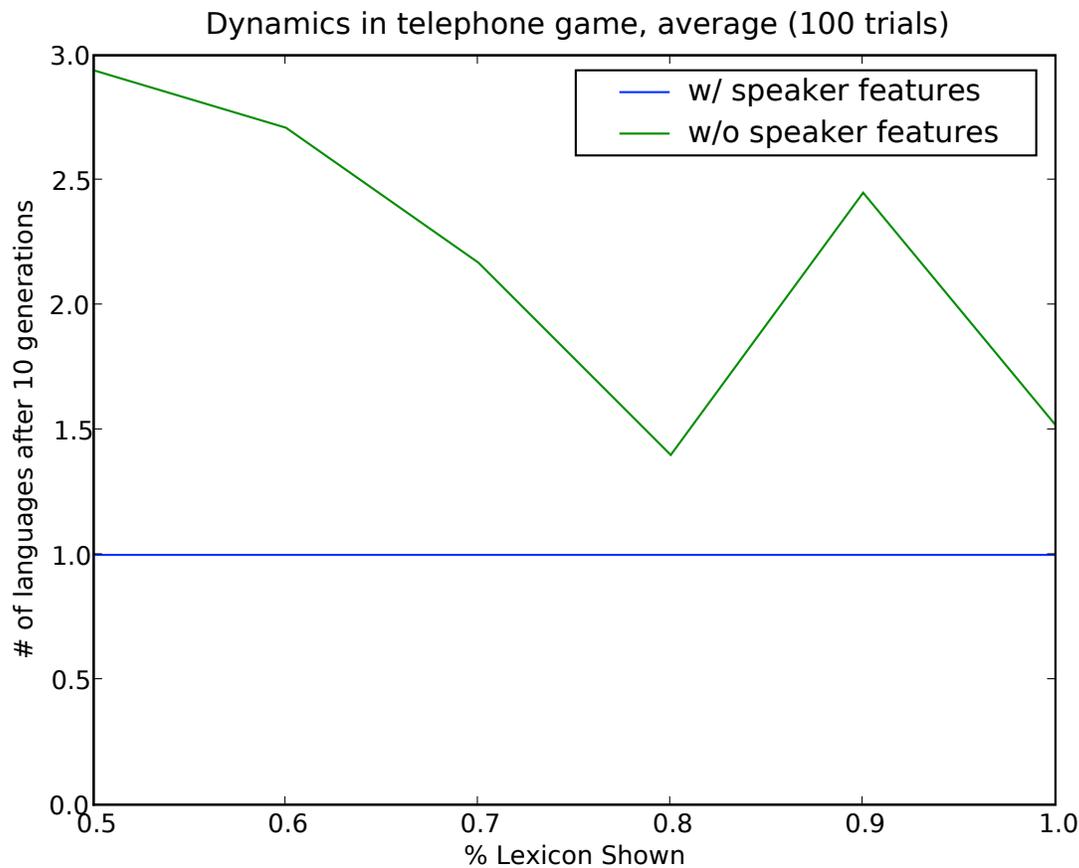

Figure 4: Average results for a series of 100 "telephone" simulations. Generation 1 begins with 1 grammar, which it uses to generate the training sample for the subsequent generation, and so on. Without the use of speaker features, the original language can quickly bifurcate into several over the generations, depending on the comprehensiveness of the sample the leaners receive.

6   Future Work

- Simulations still to be explored: Hypothesis merging *vs* speaker features.

- Some assumptions to whittle at:

  1. Supervised learning.
  2. Positive examples only.
  3. Segments, features, symbols already identified in data.
  4. No noise.
  5. One lexicon, flat usage distribution.
  6. Equal representation of target grammars in training.

7. No extralinguistic information *vs* speaker features.

8. Linearization/Maturation.

9. Strict dichotomoy between teachers, learners

10. "Telephone" population dynamics.

- Weighted co-occurrence graphs and probabilistic clustering techniques (spectral, Bayesian, etc.)

References

Biber, D. (1995). *Dimensions of Register Variation*. Cambridge University Press.

Boersma, P. & B. Hayes (2001). Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32, pp. 45–86.

Clopper, C. G. (2004). *Linguistic Experience and the Perceptual Classification of Dialect Variation*. Ph.D. thesis, Indiana University.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry* 40, pp. 35–41.

Prince, A. (2002). Entailed ranking arguments. *Rutgers Optimality Archive* ROA-500.

Prince, A. & P. Smolensky (1993). Optimality theory: Constraint interaction in generative grammar. Ms., Rutgers University and University of Colorado, Boulder.

Riggle, J. (2004). *Generation, Recognition, and Learning in Finite State Optimality Theory*. Ph.D. thesis, UCLA.

Tesar, B. & P. Smolensky (2000). *Learning in Optimality Theory*. MIT Press.