
The Power of Objects in Morphology Learning

Constantine Lignos, Jana Beck
University of Pennsylvania

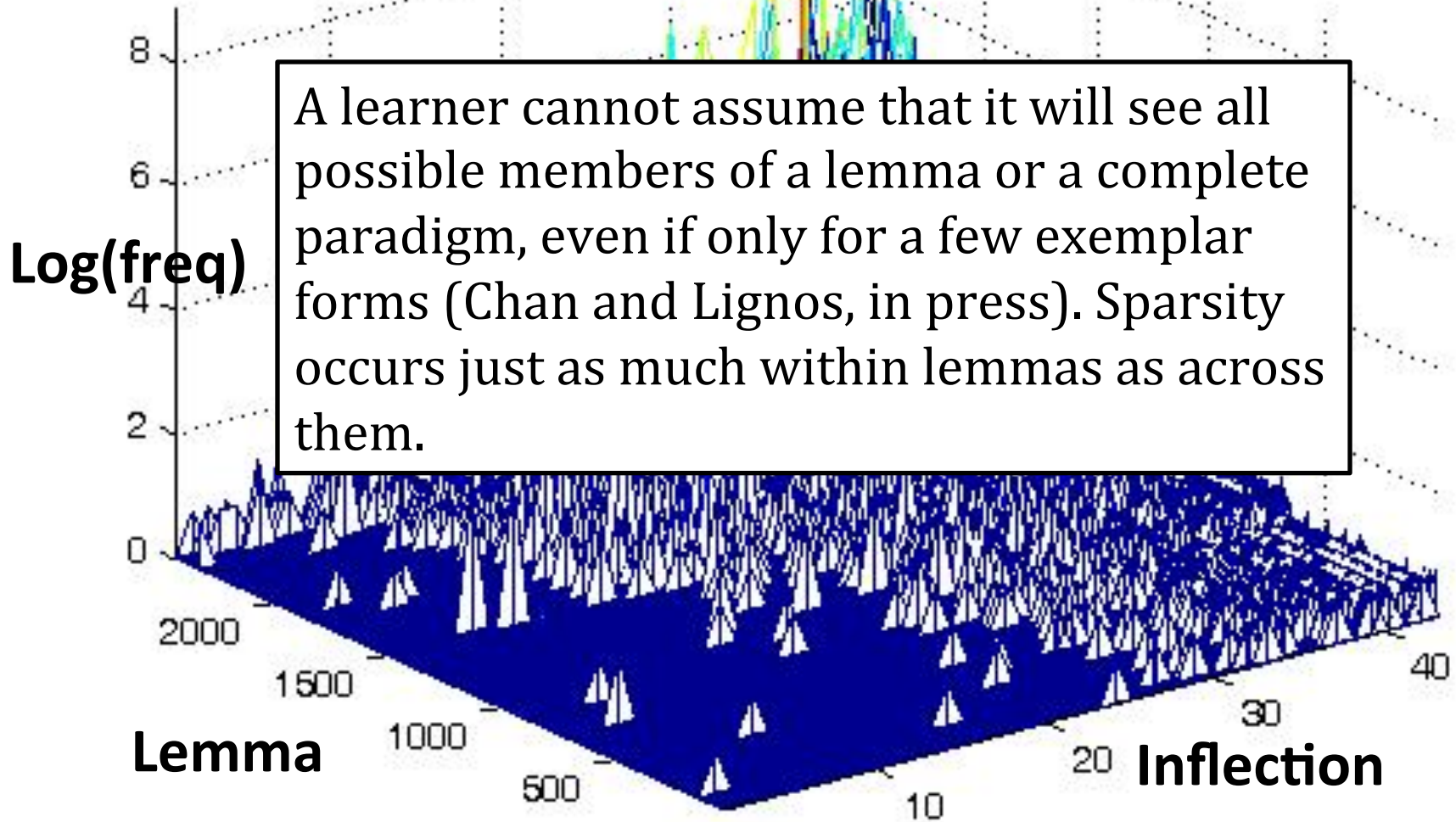
LSA Annual Meeting 2011

1/7/2011

Overview

- **We'll talk about the intersection of three areas:**
 1. Sparsity within morphological forms
 2. Simulations of unsupervised morphology learning
 3. Representations of morphology
- **We'll discuss:**
 - How sparsity presents a problem for learning models
 - How selecting the right objects (representation) can help the learner tolerate sparsity better
- **We propose that continued research at this intersection can start to bring together work in learning models, morphological theory, and processing experiments**

Sparsity in morphology (Chan, 2008)

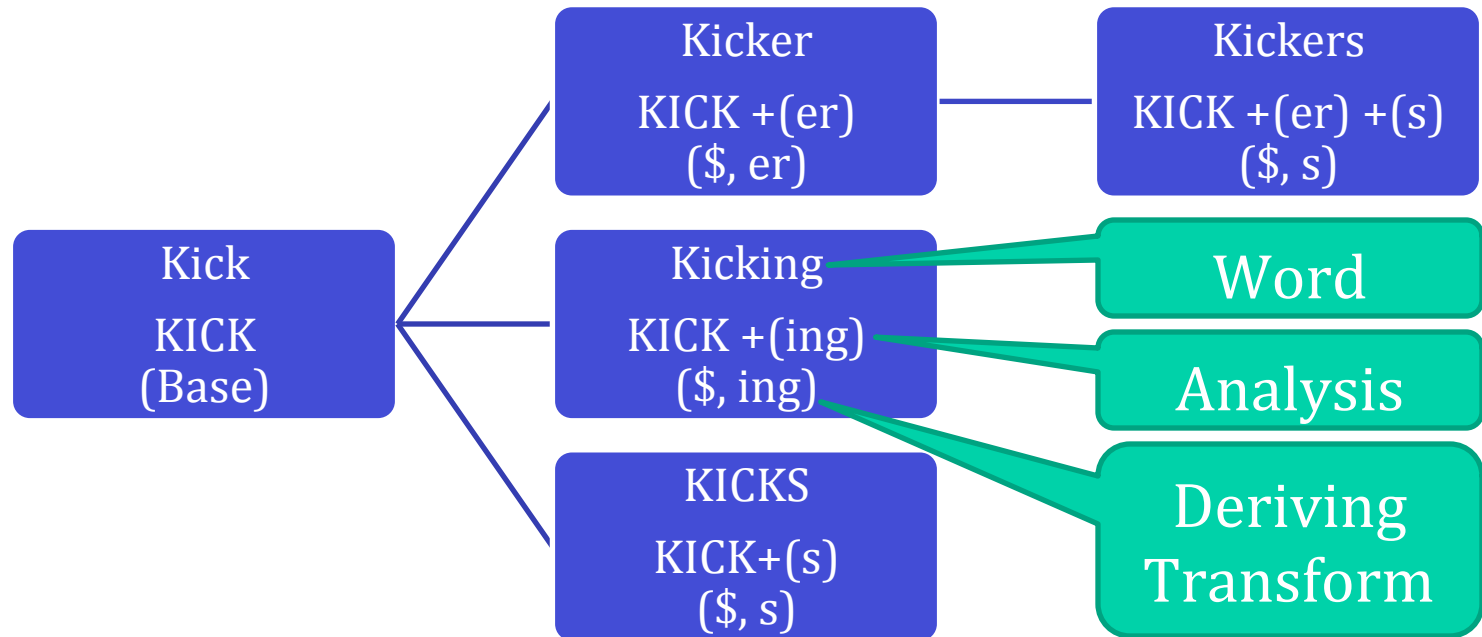


Spanish newswire verbs (2.5 M)

A simple, generative learning model

Base and Transforms Model (Chan, 2008)

- The learner will learn a set of rules (transforms) and the word pairs they apply to (base-derived pairs)



A rule learning algorithm (Chan, 2008)

- **Each iteration:**

1. Count the most type-frequent final n characters in the existing bases and the unmodeled words
 - Ex: \$, -s, -d, -r, -er, -es, -g, -ng, -ing...
2. Hypothesize transforms between the top affixes, and count how many word pairs each transform accounts for
 - Ex: (\$, s), (s, \$), (\$, ng), (ng, \$), (s, ng), (ng, s)...
3. Choose the transform that accounts for the largest number of word pairs
 - Ex: (\$, s): dog/dogs, frog/frogs, run/runs...
4. Mark the transform's base words as bases and the words it derives as modeled
 - Ex: dog + (\$, s) = dogs results in *dog* marked as a base, *dogs* marked as modeled

Structuring the lexicon

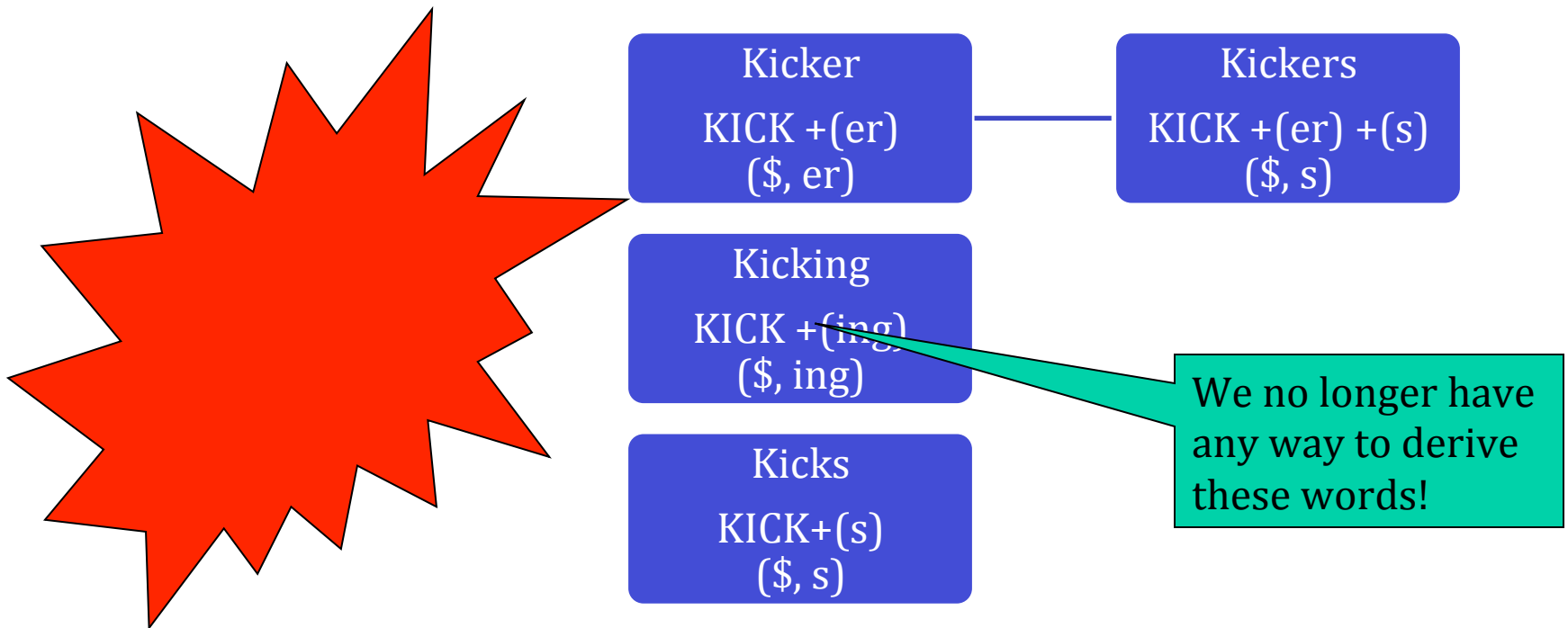
- **This model seeks structure in the lexicon through word relations similar to a number of approaches:**
 - Redundancy rules (Jackendoff, 1975)
 - Word Formation Rules (WFRs) (Aronoff, 1976, 1994; Anderson 1992)
- **Like most work concerned with morphological theory, those approaches did not develop an explicit learning model**
 - But did have a general goal of reducing the amount of information required in the lexicon

But where do bases come from? The answer isn't always trivial.

Let's consider the case of an “accidental” (sparsity-induced) missing base.

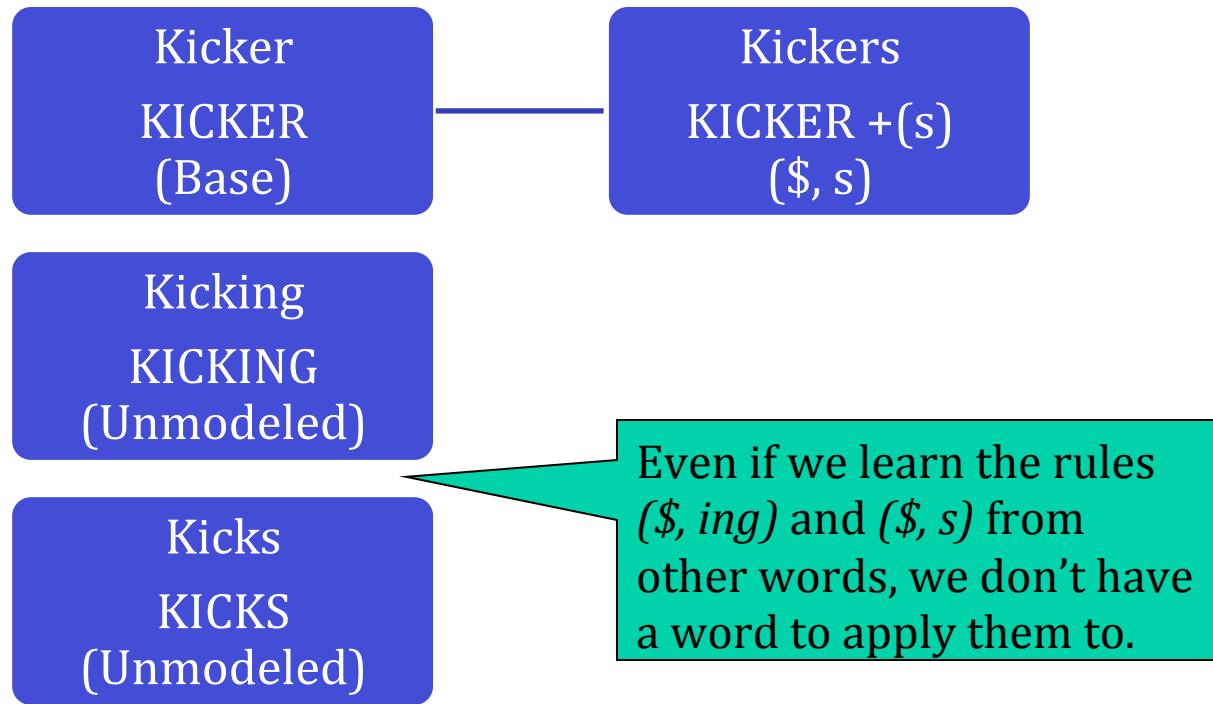
The perils of minimal pairs

- What happens to the algorithm if *kick* isn't in the input?



The new (uglier) picture

- Without the right base for the derivation chain, even if the relevant rules are learned, the desired relationships are not



It's not just a toy problem

- **While base forms are generally more frequent than forms derived from them (Wicentowski, 2002; Chan, 2008), they still are frequently unobserved**
 - Brown corpus: per CELEX analysis, 16.7% of derived words are missing their base
- **For rarer derivational forms, unobserved bases can prevent a rule from being learned at all**
- **The focus in the past has been on how to handle “intentional” gaps (**pot/potable*, **fiss/fission*) of this sort, but not accidental ones**

Working toward a solution

- The answer in the generative tradition is to learn that words whose bases are missing still have the expected structure, but how do we reliably infer that structure?
- **Naïve approach: if something could possibly be the result of applying a rule, just analyze it that way**
 - For example, we know the rule (*\$, ing*), so just assume it applies to *kicking* because it ends in *-ing*
 - For an algorithm applying this blindly is far from a good idea (*string, herring, etc.*)
- **Doing this analysis in a non-naïve way is the focus of most statistical morphology learning work, but that work provides little or no information about representation**

**A modest solution: infer the bases
you've never seen from a number of
their likely derived forms.**

Base Inference

- **To handle missing bases, infer them based on multiple words suggesting they should be there**
 - For example, in the Brown corpus *adjoins*, *adjoined*, and *adjoining* are in the corpus while *adjoin* is not
 - Allow these words to show us that *adjoin* should be in the lexicon, allowing us to *infer the missing base*
- **This inference happens as a part of the learning process, and not as a post-processing step, allowing it to feed later learning**
- **We leave aside the question of how the learner decides whether *adjoin* itself is an intentional or accidental gap**
 - i.e., whether *adjoin* is [- Lexical Insertion] (Halle, 1973)

Base Inference

Base	Derived	Unmodeled	Transforms
		Plant Plants Planted Planting Adjoins Adjoined Adjoining	

Note *adjoin* is not in the corpus.

Base Inference

Base	Derived	Unmodeled	Transforms
Plant	Plants	Planted Planting Adjoins Adjoined Adjoining	(\$, s)

Now that we learned this rule, *adjoins* suggests that *adjoin* exists, but we don't take action yet.

Base Inference

Base	Derived	Unmodeled	Transforms
Plant	Plants Planted	Planting Adjoins Adjoined Adjoining	(\$, s) (\$, ed)

Now both *adjoins* and *adjoined* suggest that *adjoin* exists. Let's pretend we saw *adjoin* and then reevaluate.

Base Inference

Base	Derived	Unmodeled	Transforms
Plant	Plants Planted		(\$, s) (\$, ed)
Adjoin	Adjoins Adjoined	Planting Adjoining	

We infer the existence of *adjoin*, and treat it as if it was there all along.

Base Inference

Base	Derived	Unmodeled	Transforms
Plant	Plants Planted Planting		(\$, s) (\$, ed) (\$, ing)
Adjoin	Adjoins Adjoined Adjoining		

Now we can apply new rules to *adjoin* just like we can to *plant*, and *adjoin/adjoining* counts toward the score of *(\$, ing)*

Evaluating Base Inference

- **Does Base Inference infer the right forms?**
 - We evaluated the precision of relationships formed by Base Inference, such as *adjoins/adjoined*
 - As evaluated using CELEX on 1 million word corpora in each language, the relationships inferred are highly precise (English, 93.5%; German, 94.9%)
- **Some errors according to the gold standard:**

integration/integrity	unemployed/unemployment
unthinkable/unthinking	automatic/automation
tune/tunic	impassable/impasse
multiplication/multiplicity	unbelievable/unbelieving
complied/compliment	rudder/ruddy

**What does this say about the
objects in the system?**

**What does the learner's
representation need to provide?**

Defining our objects

- We can now consider the question of what we want to reify—what are the *objects* in the system and what can they do?
- We take an object to mean something the learner has an explicit representation of
- To perform Base Inference correctly, these objects need to support two operations:
 - Identifying the presence of a morpheme in the output
 - s, -ed, -ing*, etc. “trigger” the Base Inference mechanism
 - In a more refined model this is more than a string match
 - Reversing/decomposing the process that generated that morpheme to create the missing base

Whither paradigms?

- **Note that while the learner benefited from having a set of rules, there are no paradigms in the representation**
 - A paradigm is not an *object* to the learner but rather an epiphenomenon of the learning *process*
- **A confluence of rules results in a viable alternative to the sparsity issues (Chan, 2008) encountered by paradigm learning approaches (Goldsmith, 2001, 2006; Monson 2008)**

Getting more from WFRs

- **Base Inference touches on a second role for WFRs, analysis for words that have no base such as *potable* (Jackendoff, 1975; Anderson, 1992)**
 - It is stated that WFRs/redundancy must have a parsing capability to cover intentional gaps (*potable*)
- **The prevalence of these missing forms suggests that this reversibility can play a crucial role to the learner outside of rarer intentional gaps—the more common case for this process is combating sparsity**
 - But we still need a more aggressive approach—only about 1 in 8 bases missing in the Brown corpus are recoverable by an ideal Base Inference process

Morphemes as objects

- **Piece-based approaches (Sciullo and Williams, 1987; Halle and Marantz, 1993) take the contrasting view that morphemes exist as more than rules that realize them**
- **The decompositional view taken by these approaches naturally integrates a process like Base Inference**
- **Building a new system from a piece-based approach we can explore the potential advantages of such a system**
 - And align with processing research on decomposition and recombination during processing (Caramazza et al., 1988; New et al., 2004; Taft, 2004; Lehtonen et al., 2006; Marslen-Wilson et al., 2008)

Closing thoughts

- **Sparsity within related forms is worse than you might think, even in English**
- **The best way to combat sparsity is to pick the right objects**
 - These objects need to be able to handle accidental gaps, which can be achieved by focusing on the reversal of rules or decomposition of morphemes
- **By building computational models that use these objects, we can compare the representations involved in morphological theories and address learning issues**

Thanks!

Many thanks to Dave Embick, Rolf Noyer, Kyle Gorman, and Charles Yang for their feedback on this work.