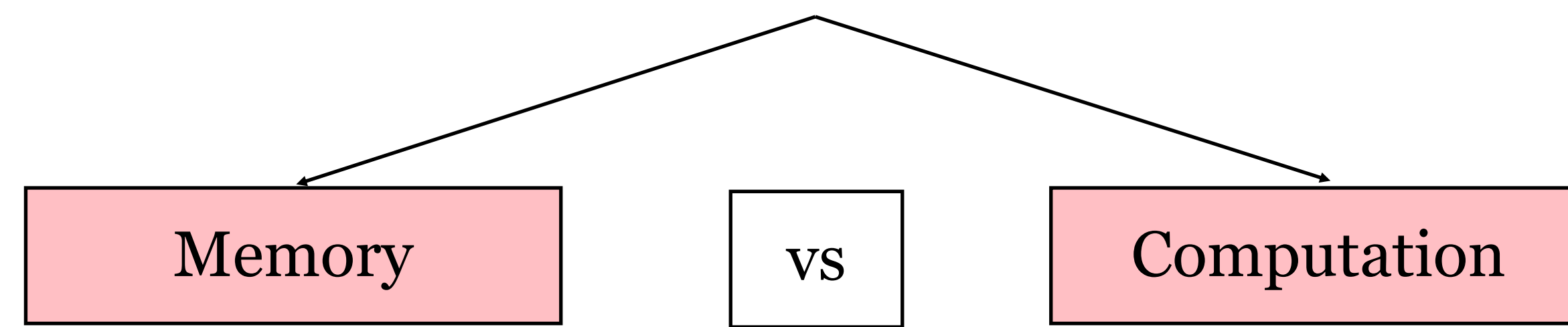


# Frequency vs. Association

## For Constraint Selection in Usage-Based Construction Grammar

### Modeling Emergence

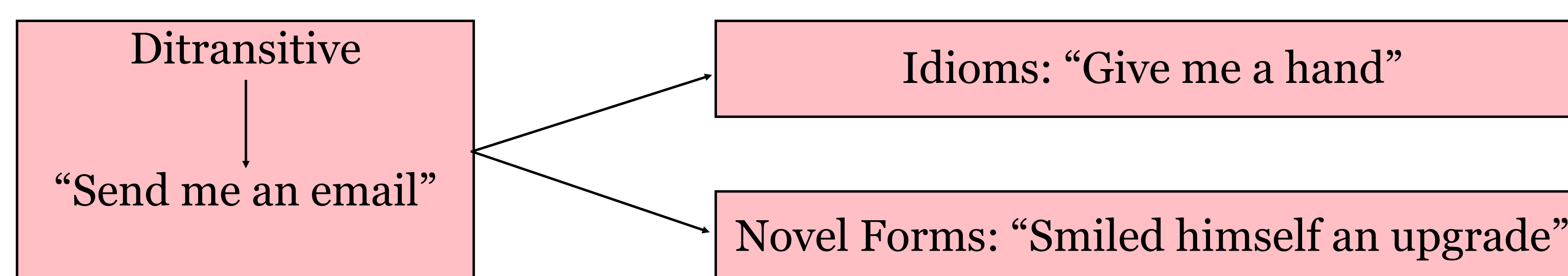
**Idea 1. Usage-based Grammar:** Any representation can be stored... but not all are worth storing



#### Question

Using a metric based on Minimum Description Length (memory vs. computation), is the best model of the generalization of constraints within constructions based on Frequency measures or Association measures?

**Idea 2. Exemplar Theory:** Grammaticalized representations emerge from exemplar / proto-type constructions



### Frequency

#### Hypothesis

A construction is a template which each slot-constraint must fit.

Search for the candidates with the highest global frequency  
(but use local association to reduce the number of candidates to count)

#### Construction-as-Template

Frequency of templates matters the most, regardless of internal relationships between slots

	1	2	3	4
LEX	"he"	"mailed"	"George"	"a package"
SYN	Noun	Verb	Noun	Noun
SEM-SYN	ANIMATE[N]	TRANSFER[V]	PERSON[N]	OBJECT[N]

	1	2	3	4
LEX	"he"	"gave"	"George"	"a hand"
SYN	Noun	Verb	Noun	Noun
SEM-SYN	ANIMATE[N]	TRANSFER[V]	PERSON[N]	OBJECT[N]

$\Delta P$  is a bi-directional measure (unlike PMI)

$$\Delta P_{LR} = p(X_P|Y_P) - p(X_P|Y_A)$$

$$\Delta P_{RL} = p(Y_P|X_P) - p(Y_P|X_A)$$

#### Variables

*line* = sequence of units  
*unit* = possible slot-constraints: (lex, syn, sem)  
 $u_i, u_{i+1}$  = two adjacent units  
 $c_i, c_{i+1}$  = constraint types for  $u_i, u_{i+1}$   
*RS* = one slot-constraint per unit in line

#### Algorithm

while *RS* not complete:  
for  $u_i, u_{i+1}$  in line:  
for all possible transitions  $c_i, c_{i+1}$ :  
if  $\Delta P(c_i, c_{i+1})$  is highest available:  
add  $c_i, c_{i+1}$  to *RS*

Table 4: Frequency-Based Selection Algorithm

### Association

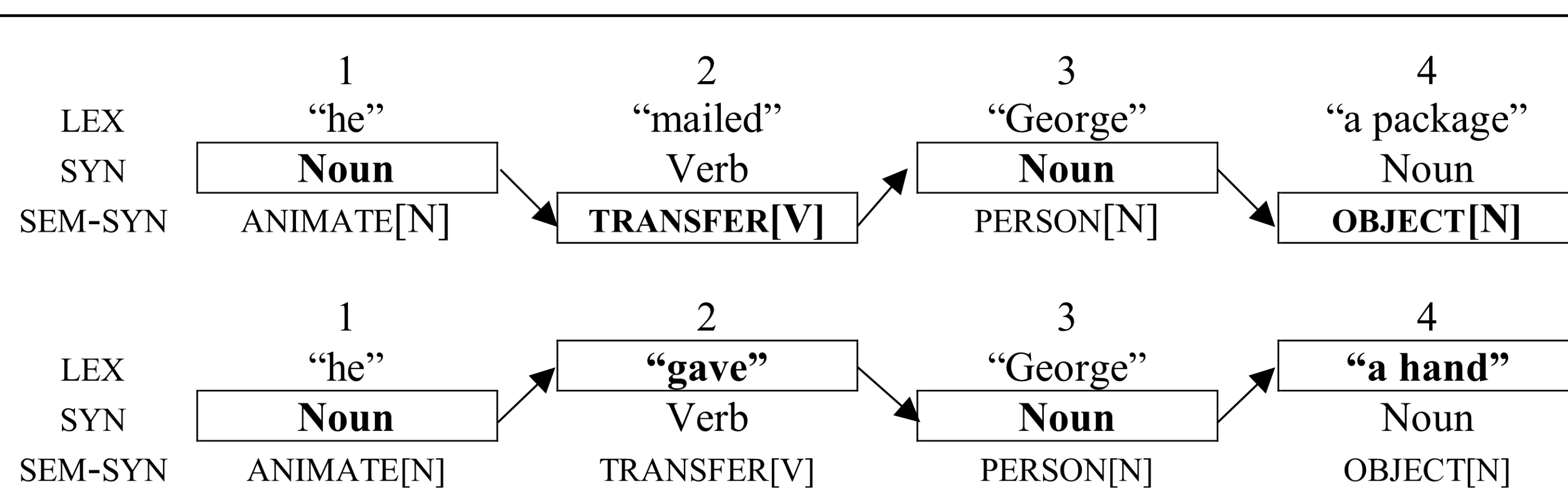
#### Hypothesis

An entrenched construction creates a chain of associated slot-constraints.

Search for the chain with the highest global association strength  
(but use global frequency as a final selection parameter)

#### Construction-as-Transitions

The best global chain of transitions outweighs frequent sequences, allowing uncommon constructions



#### Variables

*node* = unit (i.e., word) in line  
*startingNode* = start of potential construction  
*state* = type of slot-constraint for node  
*path* = route from root to successor states  
*[c]* = list of immediate successor states  
 $c_i, c_{i+1}$  = transition to successor constraint  
*candidateStack* = plausible constructions  
*evaluate* = maximize  $\sum \Delta P$  for  $c_i, c_{i+1}$  in *path*

#### Main Loop

for each possible *startingNode* in line:  
RecursiveSearch(*path* = *startingNode*)  
evaluate *candidateStack*

#### Recursive Function

RecursiveSearch(*path*):  
for  $c_i, c_{i+1}$  in *[c]* from *path*:  
if  $\Delta P$  of  $c_i, c_{i+1}$  > threshold:  
add  $c_{i+1}$  to *path*  
RecursiveSearch(*path*)  
else if *path* is long enough:  
add to *candidateStack*

Table 5: Association-Based Selection Algorithm



## Construction Grammar

1. CxG represents grammar using constraint-based *constructions* (1a and 2a)

2. Each construction is made up of slots, each of which is defined by a *constraint*

- (1a) [SYN:NOUN — SEM-SYN:TRANSFER[V] — SEM-SYN:ANIMATE[N] — SYN:NOUN]  
 (1b) “He gave Bill coffee.”  
 (1c) “He gave Bill trouble.”  
 (1d) “Bill sent him letters.”  
 (2a) [SYN:NOUN — LEX:“give” — SEM-SYN:ANIMATE[N] — LEX:“a hand”]  
 (2b) “Bill gave me a hand.”

3. Constraints are drawn from lexical, syntactic, and semantic representations

### Lexical

Word-forms from background corpus  
 (500 token threshold in ~ 1 billion words)

### Syntactic

Categories from the Universal POS tagset  
 Annotated using RDRpostagger

### Semantic

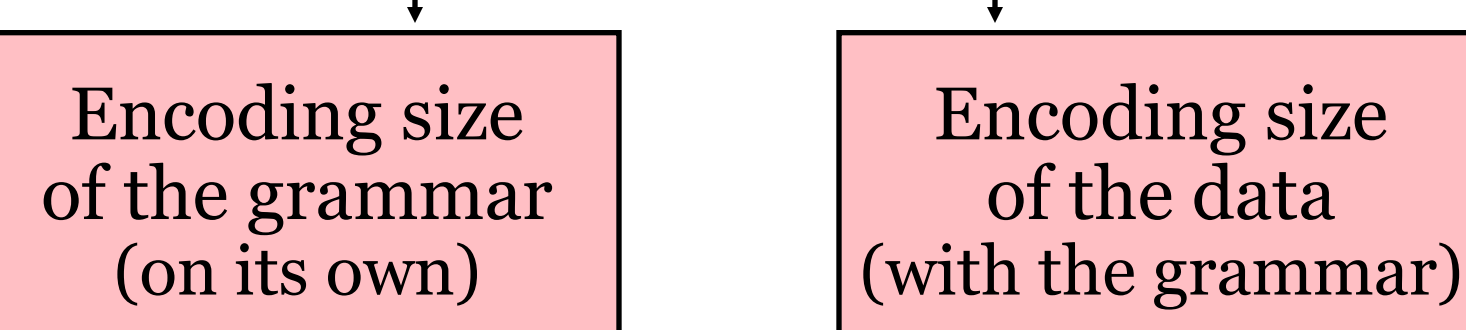
Word embeddings clustered using x-means  
 Clusters divided again by syntactic categories

## Grammar Quality

### Minimum Description Length

Operationalizes usage-based grammar’s balance between

*memory* and *computation*



$$MDL = \min_G \{L_1(G) + L_2(D | G)\}$$

Encoding size is based on probability  $L_C(X) = -\log_2 P(X)$

### Probability is Key to MDL

- Representation Types:** Considered equally probable (no explicit bias)
- Slot-Constraints:** Equally probable by type (favors smaller alphabets)
- Constructions (in L1):** Sum of representation types and constraints
- Constructions (in L2):** Based on observed frequency in training data
- Regret (in L2):** Based on frequency of unencoded words (errors)

## Results

Compression = MDL Score / Baseline  
 (lower is better)

Association-based model is significantly better on all languages

	Frequency	Association	P
ara	44.08%	<b>29.45%</b>	0.0001
deu	52.49%	<b>18.69%</b>	0.0001
eng	51.80%	<b>23.11%</b>	0.0001
fra	43.28%	<b>40.52%</b>	0.0037
por	45.13%	<b>38.91%</b>	0.0137
rus	54.14%	<b>13.93%</b>	0.0001
spa	60.34%	<b>26.36%</b>	0.0001
zho	57.01%	<b>37.96%</b>	0.0030

Table 6: Compression Rates by Language with Significance of Difference Between Models

**Experimental Set-up:** Same pipeline for both models (only selection algorithm differs) (see paper)

**Evaluation:** Calculate MDL metric on 5 independent test sets per language (each with 10 mil words)

But it is not quite so simple....

	Size of the grammar		Size of the data		Size of errors	
	$L_1(F)$	$L_1(\Delta P)$	$L_2\{C\}(F)$	$L_2\{C\}(\Delta P)$	$L_2\{R\}(F)$	$L_2\{R\}(\Delta P)$
ara	0.43%	1.25%	82.14%	68.65%	17.43%	30.10%
deu	0.50%	1.56%	89.32%	93.42%	10.17%	05.01%
eng	0.57%	1.44%	93.22%	98.04%	06.21%	00.53%
fra	0.44%	0.77%	93.08%	64.09%	06.48%	35.14%
por	0.39%	0.27%	96.72%	25.00%	02.89%	74.73%
rus	0.42%	1.35%	66.37%	94.87%	33.21%	03.78%
spa	0.36%	0.81%	99.59%	82.24%	00.06%	16.95%
zho	0.25%	0.37%	92.24%	96.92%	07.51%	02.71%

For Portuguese, errors make up most of the encoding size

Table 7: Break-down of MDL metric by relative proportion of the overall score