MDL and the complexity of natural language

John Goldsmith
University of Chicago/CNRS MoDyCo
January 2007

Thanks

- Carl de Marcken, Partha Niyogi, Antonio Galves, Jesus Garcia, Yu Hu…

The *word segmentation problem*

Input: noprincípioeraaquelequeéapalavra

Language-independent device

Output: no princípio era aquele que é a palavra

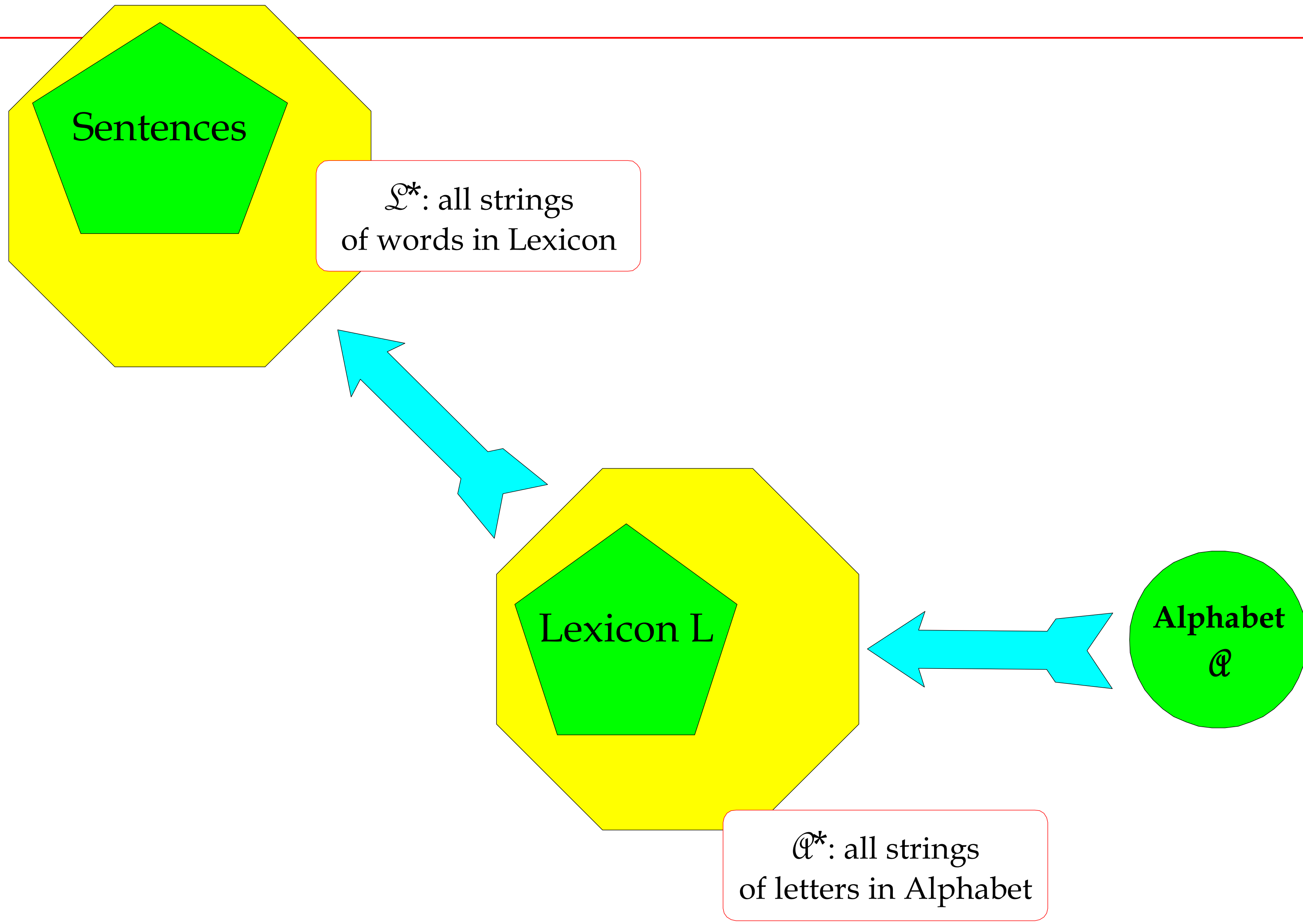
Naïve model of language

There exists an alphabet $A = \{a \ldots z\}$, and a finite lexicon $W \subset A^*$, where A^* is the set of all strings of elements of A.

There exist a (potentially unbounded) set of sentences of a language, $L \subset W^*$.

An utterance is a set (or string) of sentences, that is, an element of L^*.

Picture of naïve view



“Naïve” view?

The naïve view is still interesting –
even if it is a great simplification.

We can ask:

if we embed the naïve view inside an MDL framework, do the results resemble known words (in English, Italian, etc.)?

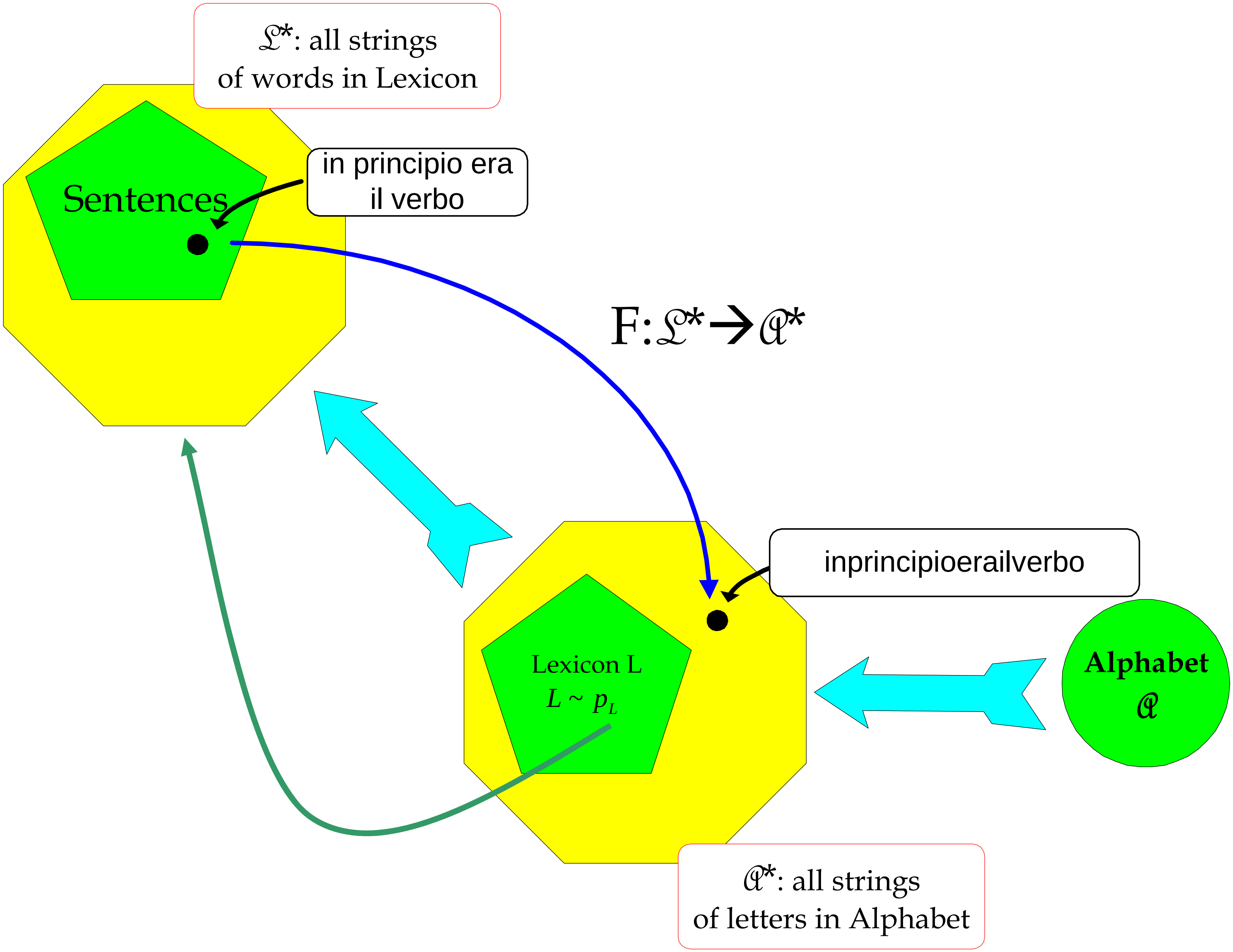
What if we apply it to DNA or protein sequences?

Word segmentation

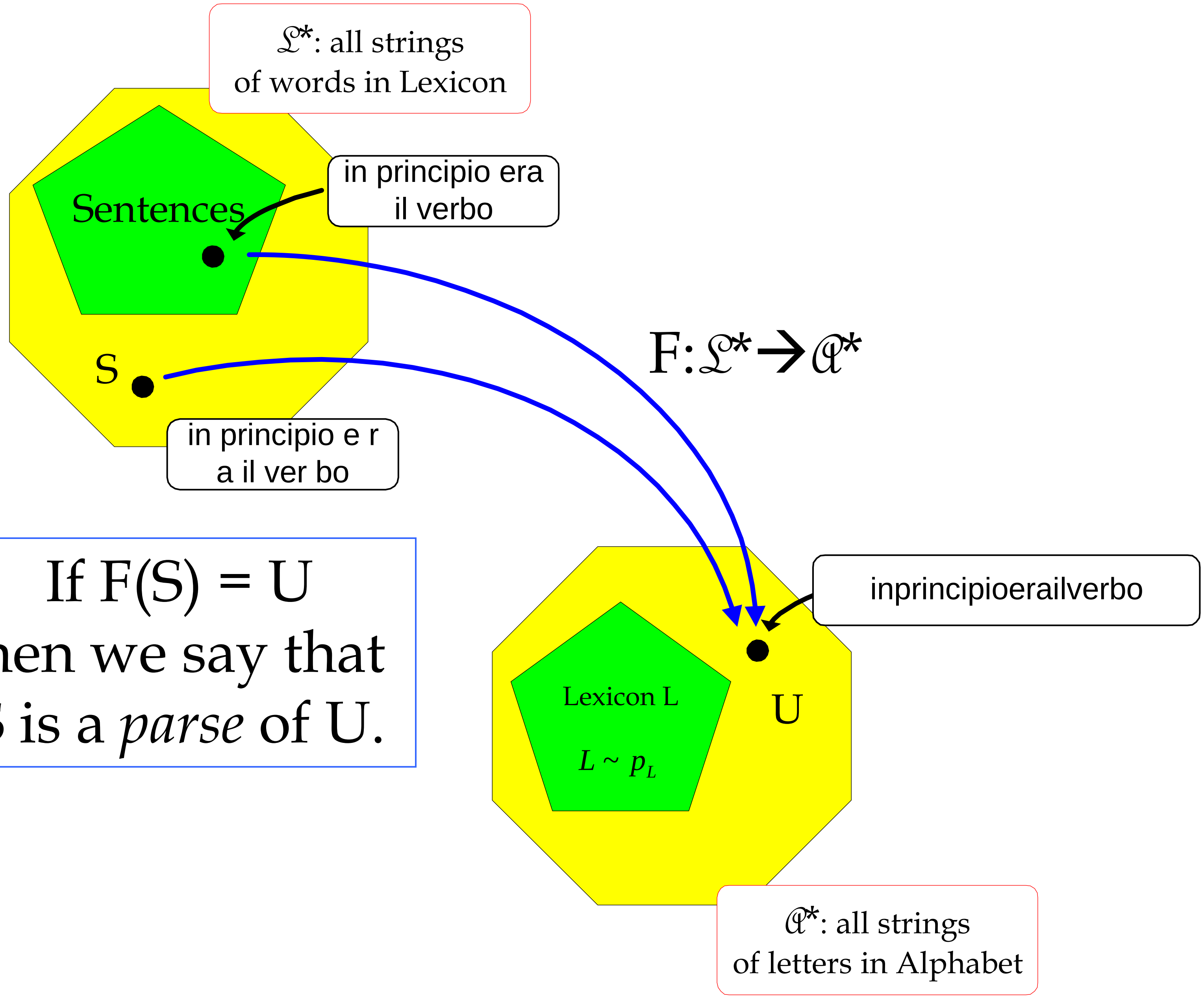
Work by Michael Brent and by Carl de Marcken in the mid-1990s at MIT.

A *lexicon* $\mathcal{L}$ is a pair of objects (L, p_L): a set L $\subset \mathcal{A}$ *, and a probability distribution p_L that is defined on $\mathcal{A}$* for which L is the support of p_L. We call L the *words*.

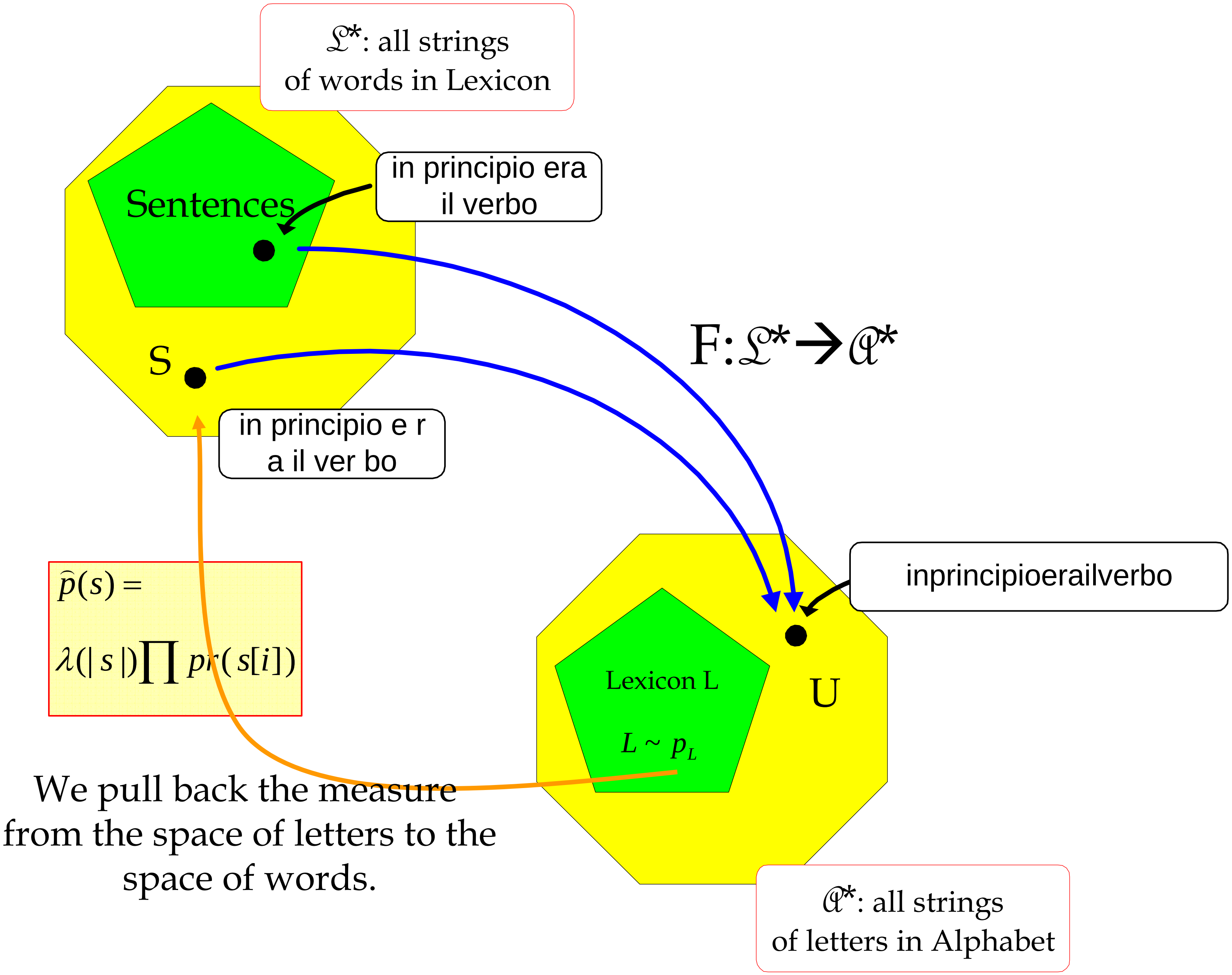
- **We insist that $\mathcal{A} \subset$ L: all individual letters are words.**
- **We define a language as a subset of L*;** its members are **sentences.**
- **Each sentence can be uniquely associated with an utterance (an element in $\mathcal{A}$ *) by a mapping F:**



ℒ*: all strings
of words in Lexicon
Sentences
in principio era
il verbo
F:ℒ*→𝒜*
inprincipioerailverbo
Lexicon L
L ~ p_L
Alphabet
𝒜
𝒜*: all strings
of letters in Alphabet



If F(S) = U
then we say that
S is a *parse* of U.



$\mathscr{L}$*: all strings
of words in Lexicon
Sentences
in principio era
il verbo
S
in principio e r
a il ver bo
F: $\mathscr{L}^* \rightarrow \mathscr{A}^*$
inprincipioerailverbo
$\hat{p}(s) =$
$\lambda(|s|)\prod pr(s[i])$
Lexicon L
$L \sim p_L$
U
$\mathscr{A}$*: all strings
of letters in Alphabet
We pull back the measure
from the space of letters to the
space of words.

Different lexicons lead to different probabilities of the data

Given an utterance U

$$p_L(U \mid L) = \underset{q \in \{parses(U)\}}{\arg\max}\ \widehat{p}_L(q)$$

The probability of a string of letters is the probability assigned to its best parse.

Class of models originally studied in the word segmentation problem

[eventually we will come to regret the limitations of this class…]

Our data is a finite string (“corpus”), generated by a finite alphabet;

We find the best parse for the string;

The probability of the parse is the product of the probability of its words;

The words are assigned a maximum likelihood probability of the simplest sort.

A little example, to fix ideas

How do these two multigram models of English compare? Why is Number 2 better?

Lexicon 1:
{a,b,…,h,…,s, t, u…z}

Lexicon 2:
{a,b,…,h,…s, t, th, u…z}

A bit of notation

Notation:

[t] = count of *t*

[h] = count of *h*

[th] = count of *th*

Z = total number of words (tokens)

$$Z = \sum_{l \in lexicon} [l]$$

Log probability of corpus:

$$\sum_{m\ in\ lexicon} [m] \log \frac{[m]}{Z}$$

$$\sum_{m\ \text{in lexicon}} [m] \log \frac{[m]}{Z}$$

$$\textit{where } Z = \sum_{l \in lexicon} [l]$$

Log prob of sentence C

$$[t]_1 \log \frac{[t]_1}{Z_1}$$

$$+ [h]_1 \log \frac{[h]_1}{Z_1}$$

$$+ \sum_{m \neq t,h} [m] \log \frac{[m]}{Z_1}$$

All letters are separate

$$[t]_2 \log \frac{[t]_2}{Z_2}$$

$$+ [h]_2 \log \frac{[h]_2}{Z_2}$$

$$+ \sum_{m \neq t,h} [m] \log \frac{[m]}{Z_2}$$

$$+ [th]_2 \log \frac{[th]_2}{Z_2}$$

th is treated as a separate chunk

$$[t]_2 = [t]_1 - [th]$$

$$[h]_2 = [h]_1 - [th]$$

$$[Z]_2 = [Z]_1 - [th]$$

$$[t]_1 \log \frac{[t]_1}{Z_1} + [h]_1 \log \frac{[h]_1}{Z_1} + \sum_{m \neq t,h} [m] \log \frac{[m]}{Z_1}$$

All letters are separate

$$[t]_2 \log \frac{[t]_2}{Z_2} + [h]_2 \log \frac{[h]_2}{Z_2} + \sum_{m \neq t,h} [m] \log \frac{[m]}{Z_2} + [th]_2 \log \frac{[th]_2}{Z_2}$$

th is treated as a separate chunk

$$define\ \Delta f\ \ as\ \log \frac{f_2}{f_1}; then\ \Delta pr(C) =$$

$$-Z_1 \Delta Z + [t]_1 \Delta t + [h]_1 \Delta h + [th] \log \frac{pr_2(th)}{pr_2(t)\, pr_2(h)}$$

This is **positive** if Lexicon 2 is better

Effect of having
fewer "words" altogether

$$define\ \Delta f\ \ as\ \log \frac{f_2}{f_1};then\ \Delta pr(C) =$$

$$\boxed{-Z_1\Delta Z + [t]_1\Delta t + [h]_1\Delta h + [th]\log \frac{pr_2(th)}{pr_2(t)\,pr_2(h)}}$$

This is **positive** if
Lexicon 2 is better

Effect of frequency
of /t/ and /h/ decreasing

$$\textit{define } \Delta f \textit{ as } \log \frac{f_2}{f_1}; \textit{then } \Delta pr(C) =$$

$$-Z_1 \Delta Z + [t]_1 \Delta t + [h]_1 \Delta h + [th] \log \frac{pr_2(th)}{pr_2(t)\, pr_2(h)}$$

This is **positive** if
Lexicon 2 is better

Effect /th/ being
treated as a unit
rather than separate pieces

$$define\ \Delta f\ \ as\ \log\frac{f_2}{f_1};then\ \Delta pr(C)=$$

$$\boxed{-Z_1\Delta Z+[t]_1\Delta t+[h]_1\Delta h+[th]\log\frac{pr_2(th)}{pr_2(t)\,pr_2(h)}}$$

This is **positive** if
Lexicon 2 is better

Description Length

We need to account for the increase in length of the Lexicon, which is our model of the data.

We add "th" to the lexicon:

$$\log \frac{Z_2}{[t]} + \log \frac{Z_2}{[h]} = -\log(pr_2(t)\,pr_2(h))$$

$$\boxed{-Z_1 \Delta Z + [t]_1 \Delta t + [h]_1 \Delta h + [th] \log \frac{pr_2(th)}{pr_2(t)\,pr_2(h)} - \log(pr_2(t)\,pr_2(h))}$$

This is the generic form of the MDL criterion for *adding* a new word to the lexicon.

Results

- The Fulton County Grand Ju ry s aid Friday an investi gation of At l anta 's recent prim ary e lection produc ed no e videnc e that any ir regul ar it i e s took place .
- Thejury further s aid in term - end present ment s thatthe City Ex ecutive Commit t e e ,which had over - all charg e ofthe e lection , d e serv e s the pra is e and than k softhe City of At l anta forthe man ner in whichthe e lection was conduc ted.

Chunks are too big Chunks are too small

Start with:

BREVES INSTRUCÇÕES AOS CORRESPONDENTES
DA ACADEMIA DAS SCIENCIAS
DE LISBOA 1781

As relações, por mais exactas e completas que sejão, nunca chegão a dar-nos huma idéa tão perfeita das coisas, como a sua mesma presença: por esta causa se tem occupado os Sabios, particularmente neste seculo, em ajuntar com a protecção dos Principes os exemplares de varios individuos das diversas especies de Animaes, Vegetaes e Mineraes, que se encontrão em differentes paizes, para apresentarem do modo possivel á vista dos curiosos hum como compendio das principaes maravilhas da Natureza.—

Remove spaces

- Asrelações,pormaisexactasecompletasquesejão,nuncachegãoadar-noshumaidéatãoperfeitadascoisas,comoasuamesmapresença:porestacausasetemoccupadoosSabios,particularmentenesteseculo,emajuntarcomaprotecçãodosPrincipesosexemplaresdevariosindividuosdasdiversasespeciesdeAnimaes,VegetaeseMineraes,queseencontrãoemdifferentespaizes,paraapresentaremdomodopossivelávistadoscuriososhumcomocompendiodasprincipaesmaravilhasdaNatureza.—

- As relações ,pormais exacta—se complet—as que sejão , nunca che—gão a da—r-nos humaidéa tão perfeita das coisas, como asu—a mes—ma-presenç—a : por esta caus—a setem occupa—do os S—abios, particula—r—mente neste seculo , em ajuntar coma prote—cção dos Principes os exemplaresde varios individuos dasdivers—asespeciesde An—imaes, Vege—ta—e—se Min—eraes,que se encontr—ãoem differentes paizes ,para apresenta—rem do modopossivel á vista dos curios-os hum como compendi—o das principa—es maravilhas da Natureza.

What do we conclude?

- From the point of view of linguistics, this does not teach us something about language (at least, not directly).
- From the point of view of statistical learning, this does not teach us about statistical learning procedures.

What do we conclude?

What is most interesting about the results is that the linguist sees the ***errors*** committed by the system (by comparison with standard spelling, e.g.) as the result of a specification of a model set which *fails to allow a method* to capture the structure that linguistics has analyzed in language.

We return to this…

…in a moment.

First, an observation the behavior of MDL in this process, so far.

Usage of MDL?

If ***description length*** of data D, given model M, is equal to

the inverse log probability assigned to D by M +

compressed length of M, then

The process of word-learning is unambiguously one of increasing the probability of the data, and using the length of M as a stopping criterion.

Discovering words from letters:

Decrease compressed length of data,

Use length of model as a stopping criterion.

Linguistic cases we will see below:

Decrease length of model,

Use data compression improvement as a stopping criterion.

$$\{(D_0, G_0), (D_1, G_1), (D_2, G_2), (D_3, G_3), \ldots (D_N, G_N), \quad \}$$

$$\| G \| = \textit{compressed length of grammar}$$

$$\| D \| = \textit{compressed length of data}$$

Subscript represents iteration in learning process

Good: $\| D_{i+1} \| < \| D_i \|$

$\| G_{i+1} \| > \| G_i \|$

$\| D_{i+1} \| > \| D_i \|$

Good: $\| G_{i+1} \| < \| G_i \|$

Conjecture

Suppose: the *data* we wish to account for is *all* of the textual data on the Internet in the world's various languages, *plus* the alignment between corresponding sentences in the case of texts appearing in more than one language.

We wish to find the minimal description of all of this data.

Conjecture

Conjecture (version 1): if we find the optimal compression, we will discover the traditional categories of linguistic analysis inside it (morphology, syntax, semantics, etc.).

Conjecture (version 2): in order to approach this optimum in a tractable fashion with an automatic learning algorithm, we need to explicitly include categories of linguistic analysis.

3 major categories of failures of naïve model of word learning:

- Many failures of word-discovery are correct discovery of morphemes (word-pieces) **investi-gation**, **complet – as**.
- Many (thought fewer) failures of word-discovery are discovery of pairs of words that frequently appear together (for example, ***ofthe***).
- Many failures are too short to be likely words.

Today's focus: #1

Finding word-internal structure and using it in the computation of description length.

Conclusion

Linguistica Project: under way since 1997 at http://linguistica.uchicago.edu

Developed to rapidly discover morphological structure in an increasingly large number of natural languages with *no prior knowledge* of the languages.

Morphology

Ask a linguist: it is *the study of word-internal structure*

Ask a statistician: it is the extraction of certain aspects of redundancy in the vocabulary of a language.

We describe a morphology analyzer (*Linguistica*) that learns morphology with *no* knowledge of the language.

In order to shrink ||G||…

There are about 74 different forms of each verb (*cantar, canto, cantas, canta, cantamos, cantais, cantam, …cantassem,…*). Each letter takes very roughly 4 bits to encode; there are a total of 576 letters ~2,300 bits.

cant- is 4 letters long; each letter takes ~4 bits to encode; hence each appearance of *cant* requires ~16 bits.

Why repeat *cant* each time?

Language allows a data structure at least this complex:

We could shrink the morphology:

$$cant\left\{\begin{array}{c} o \\ as \\ a \\ \text{...71 } more\text{...} \end{array}\right\}$$

Compared to a simple word list, we save 73 repetitions of *parl* (= *73*16 bits* = *1168 bits*), minus the price T of the data structure represented by "___{ }".

Order of magnitude

Using this data structure allows us to save roughly 1170 bits out of 2304 (51%).

How much do we have "pay" in order to encode the data structure? We called this T…

$$[stem]\left\{\begin{matrix} o \\ as \\ a \\ ...71more... \end{matrix}\right\}$$

Calculate *T*

- Notice that it’s not the cost of expressing those suffixes (that cost would have to be paid *anyway):* it’s the cost of expressing the notion “this stem may be followed be these suffixes”.
- There are hundreds of verb stems in Portuguese that will use exactly the same data structure, because they accept exactly the same suffixes.

More generally

$$\begin{Bmatrix} cant \\ lav \\ am \end{Bmatrix} \begin{Bmatrix} o \\ i \\ a \\ \text{...71}more\text{...} \end{Bmatrix}$$

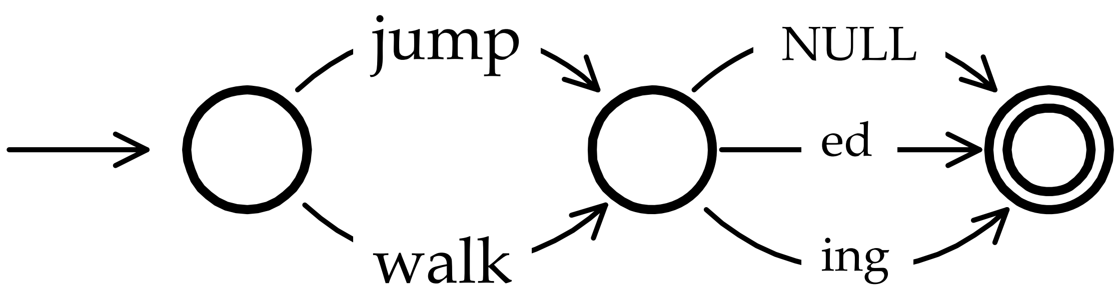
$$\begin{Bmatrix} élevé \\ équipé \\ étonnant \\ \text{78 } more \end{Bmatrix} \begin{Bmatrix} NULL \\ e \\ s \\ es \end{Bmatrix}$$

$$\begin{Bmatrix} account \\ appeal \\ attack \\ \text{40 } more\text{...} \end{Bmatrix} \begin{Bmatrix} NULL \\ ed \\ ing \end{Bmatrix}$$

- We calculate T by calculating the cost of specifying a finite state automaton with labeled edges.

Finite state automaton (FSA)

$$\left\{ \begin{matrix} jump \\ walk \end{matrix} \right\} \left\{ \begin{matrix} NULL \\ ed \\ ing \end{matrix} \right\}$$

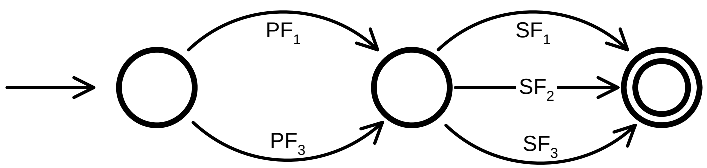


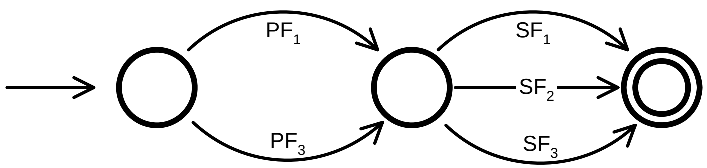
DL savings and costs

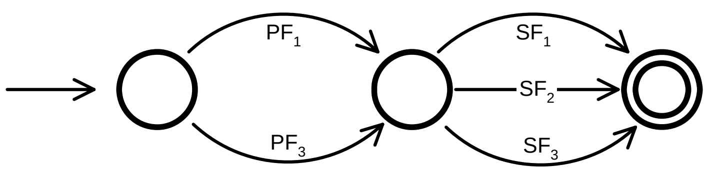
Specification of the vocabulary of a lexicon of a language by a finite state automaton can lead to considerable savings in description length.

1. We must make explicit the cost of an FSA;
2. And the change in the compression of the original data.

Cost of an FSA







For each FSA, we "pay for" the information required to specify each state, each transition, and each label of each transition.

[σ] = Number of times a signature is used in the data.

Z= size of data.

Size of pointer to first state of each signature = $\log_2 \frac{Z}{[\sigma]}$

Initial approximation

- We assume a morphology is a collection of 3 state FSAs, all sharing a unique final state.
- Then the cost is the sum of the costs of the pointers to the first states, plus the cost of labeling the edges.

Complexity of model

$$\log(|\Sigma|) + \sum_{\sigma \in \Sigma} \left(\frac{Z}{[\sigma]} + \sum_{t \in Stems(\sigma)} \frac{Z}{[t]} + \sum_{f \in Suffixes(\sigma)} \frac{Z}{[f]} \right)$$

$$+ \sum_{t \in T} |t| \log 27 + \sum_{f \in F} |f| \log 27$$

Probability of a sentence

$$pr(w) =$$

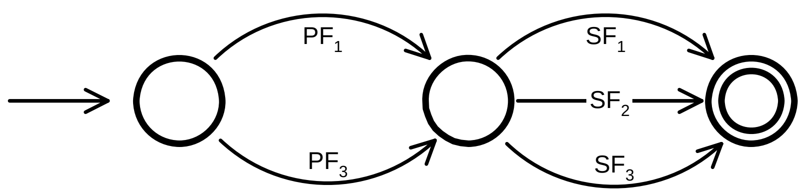
$$pr(\sigma(w))\, pr(stem \mid \sigma)\, pr(suffix \mid \sigma)$$

Log prob (corpus)

$$\log prob(corpus) =$$

$$\sum_{\sigma \in \Sigma} \left\{ \begin{array}{l} [\sigma] \log prob(\sigma) + \\ \displaystyle\sum_{t \in stems(\sigma)} [t] \log prob(t \mid \sigma) + \\ \displaystyle\sum_{f \in \sigma} [f\ in\, \sigma \mid \sigma] \log prob(f \mid \sigma) \end{array} \right\}$$

Benefits of re-using labels for affixes



There is considerable benefit to labeling the affixes *not* with strings, but with *pointers to strings.* The information cost of such a label more expensive if it is used only once, but if it is re-used a great deal, there is rapid gain to the MDL system: in short, the model demands generalizations in the grammar.

How?

Not all analyses are correct:

$$car\left\{\begin{matrix} d \\ e \\ l \\ p \end{matrix}\right\}$$

But some are:

$$act\left\{\begin{matrix} NULL \\ ed \\ s \\ ion \end{matrix}\right\}$$

- The difference lies in the very low cost associated with creating

$$act\begin{Bmatrix} NULL \\ ed \\ s \\ ion \end{Bmatrix}$$

and the relatively high cost associated with creating

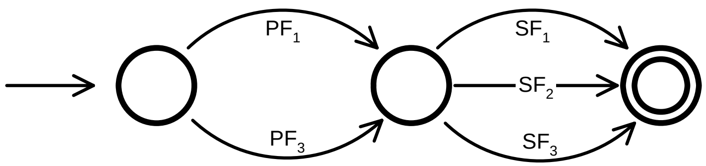
$$car\begin{Bmatrix} d \\ e \\ l \\ p \end{Bmatrix}$$

in which *l* and *p* are extremely rare (unique) suffixes: hence a pointer to each of them is very costly in bits.

Whether we think of the object this way:

$$\boxed{\left\{\begin{matrix} \textit{account} \\ \textit{appeal} \\ \textit{attack} \\ 40\ \textit{more...} \end{matrix}\right\}\left\{\begin{matrix} \textit{NULL} \\ \textit{ed} \\ \textit{ing} \end{matrix}\right\}}$$

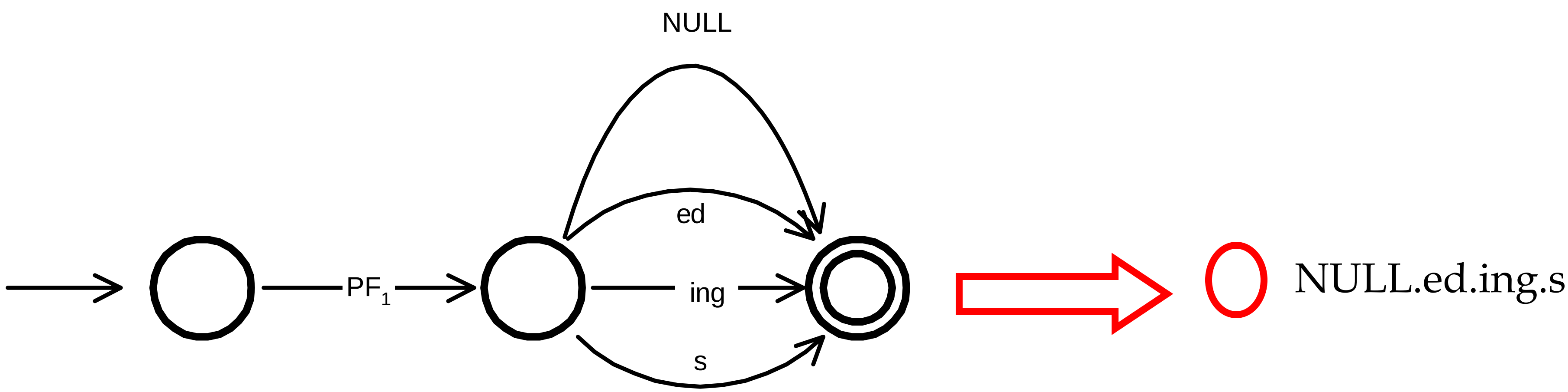
Or this way:



It is often convenient to think of it as an an abstract object.

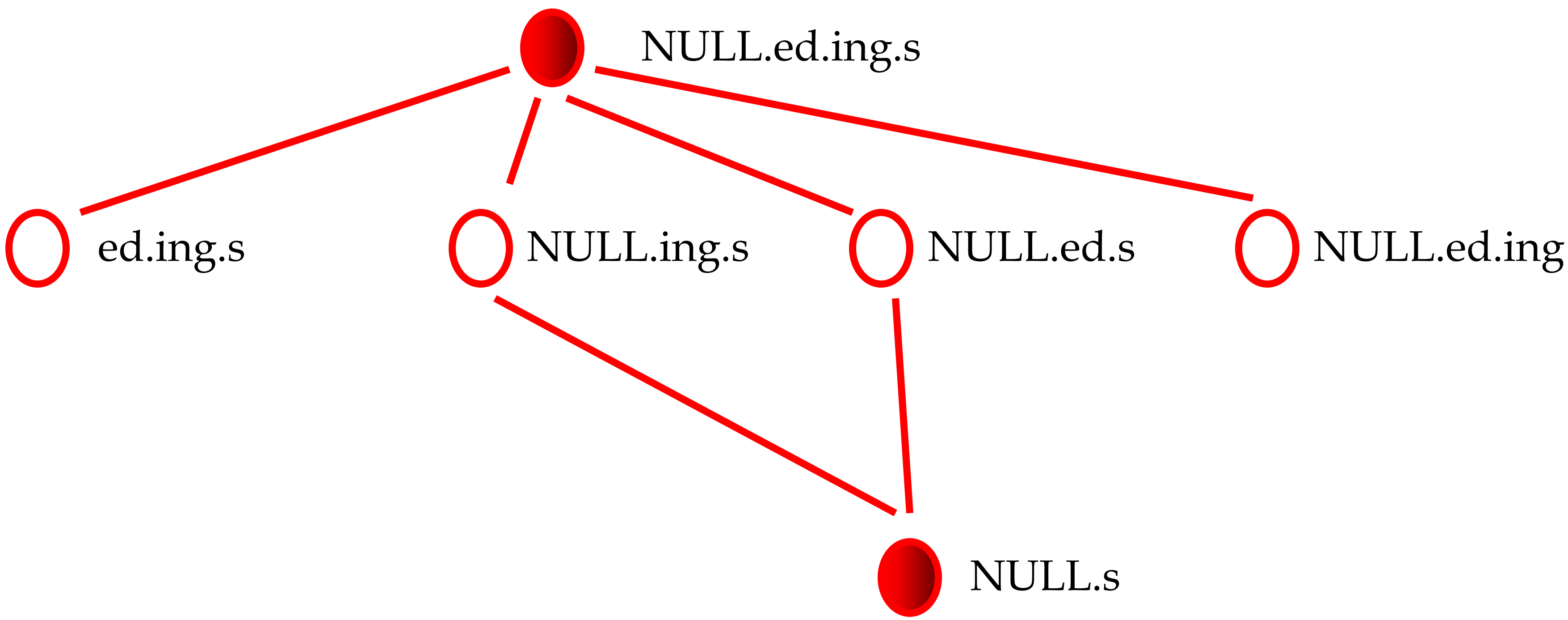
There is a natural embedding of this object into a lattice in the following sense:

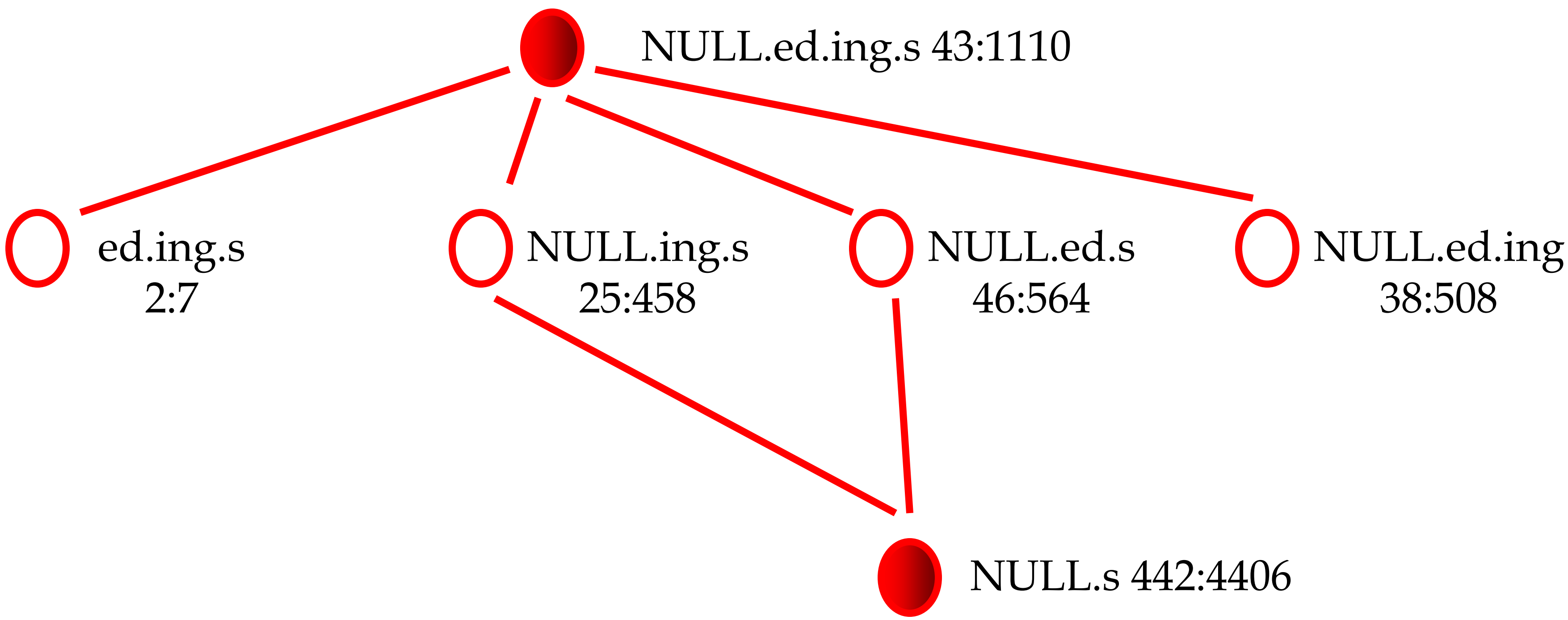
Each node is an FSA;
Each FSA is a node



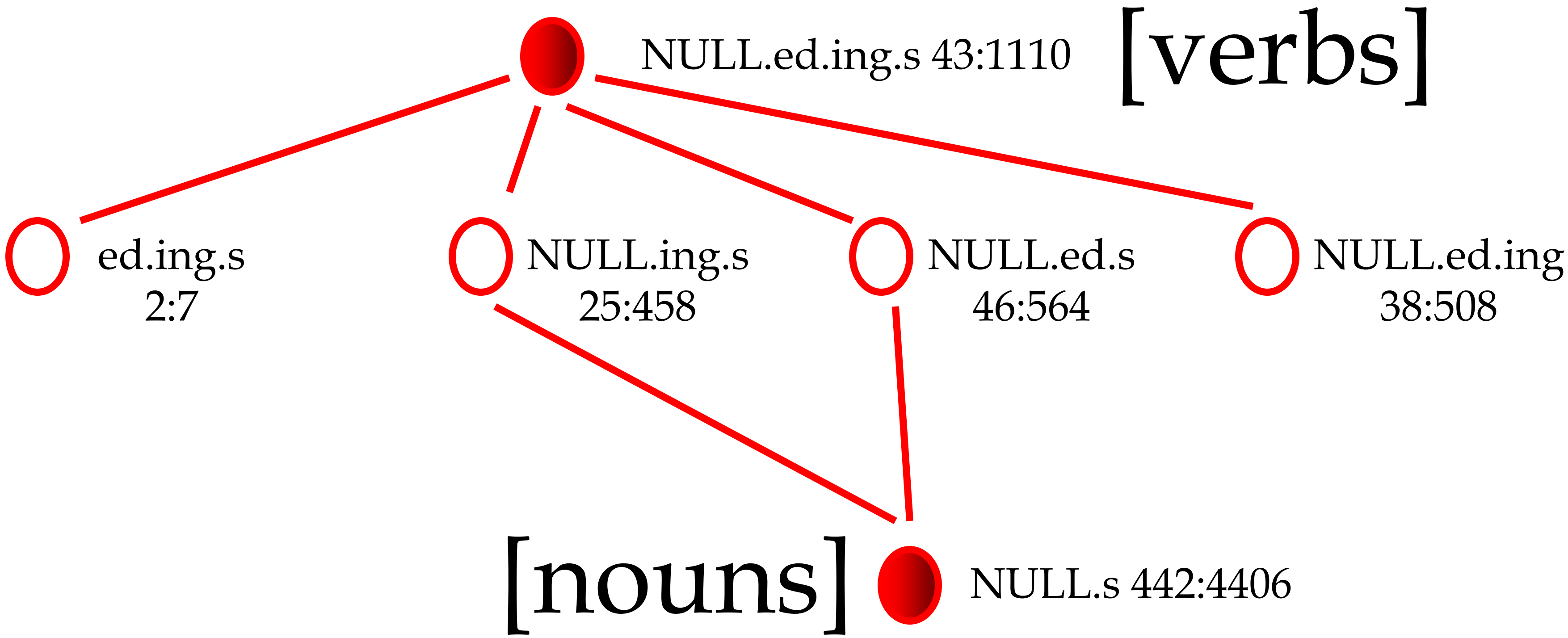
Embed the nodes in the lattice generated by the set of suffixes.

Edges represent set inclusion

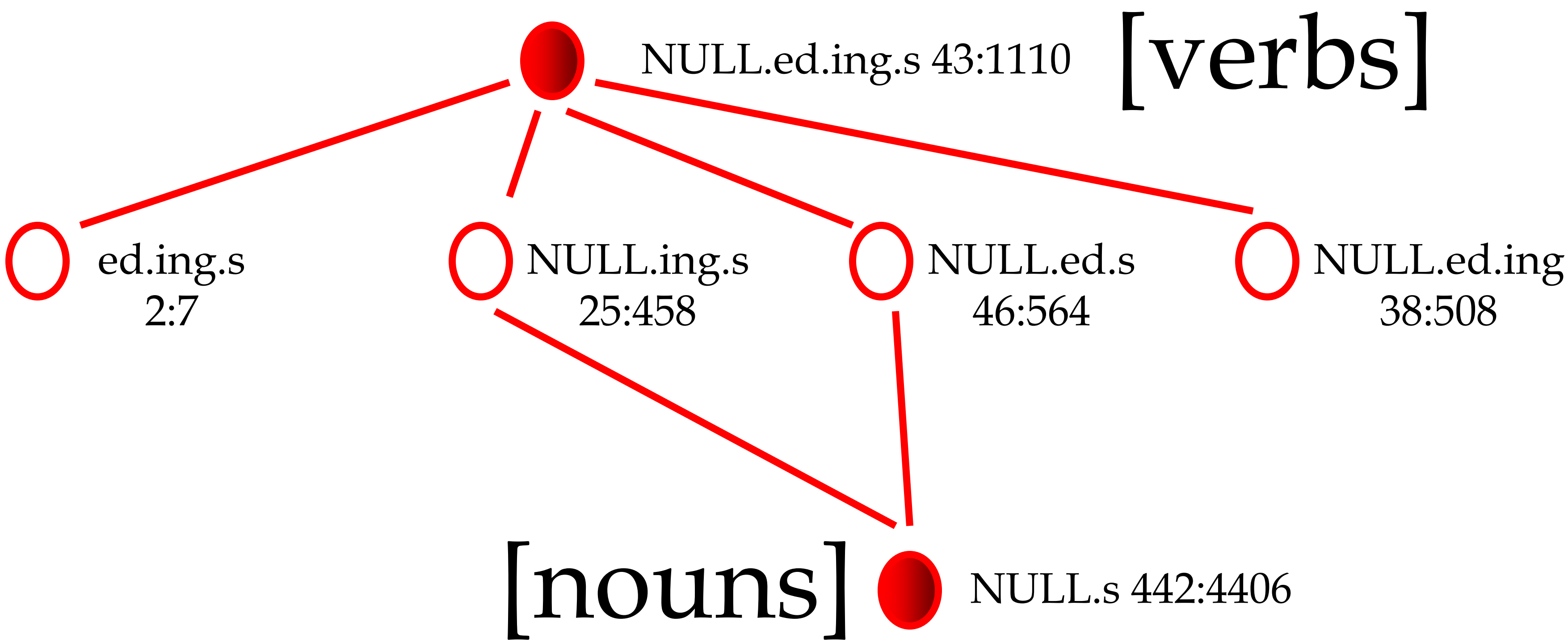




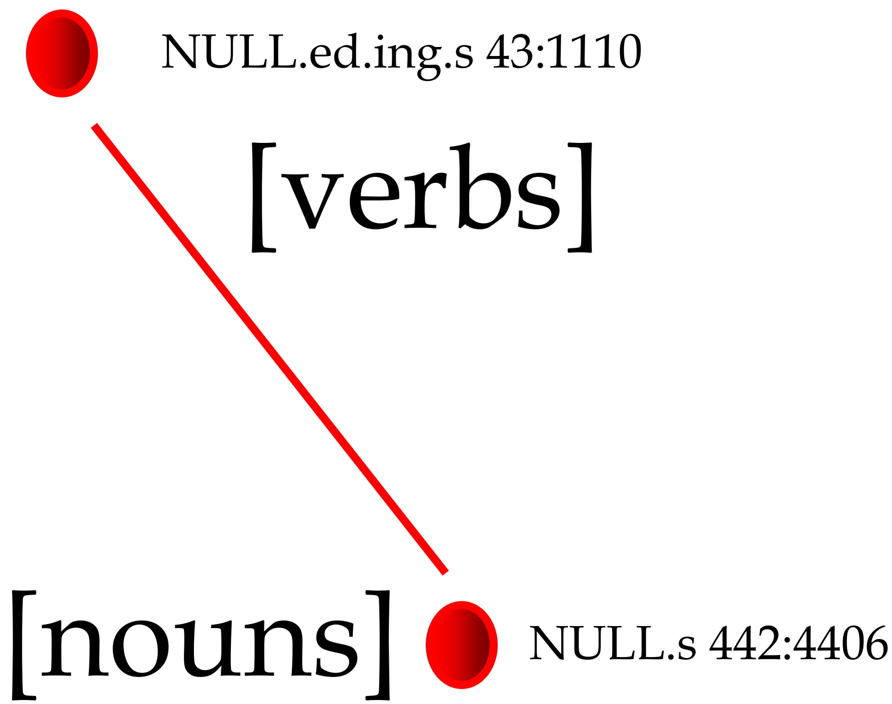
Notation:
Suffix1.Suffix2
#stems: # occurrences



Generalization
consists of eliminating nodes,
and push their stems upward



Eliminate *unsaturated* nodes, found in the data but accidental



Eliminate *unsaturated* nodes, found in the data but accidental

A glimpse of other work

The FSAs for real language data are much more complex than just a set of independent 3-state FSAs (finite state automata).

3 Questions a linguist would ask

- What is the grammar of this long sample from (Swahili/English/Italian/…): or, what is the grammar of Swahili?
- What is the nature of human language?
- What is linguistics?

3 possible answers

- What is Swahili? Find the most compact representation of the sample (the "corpus") you have.

2. What is human language?

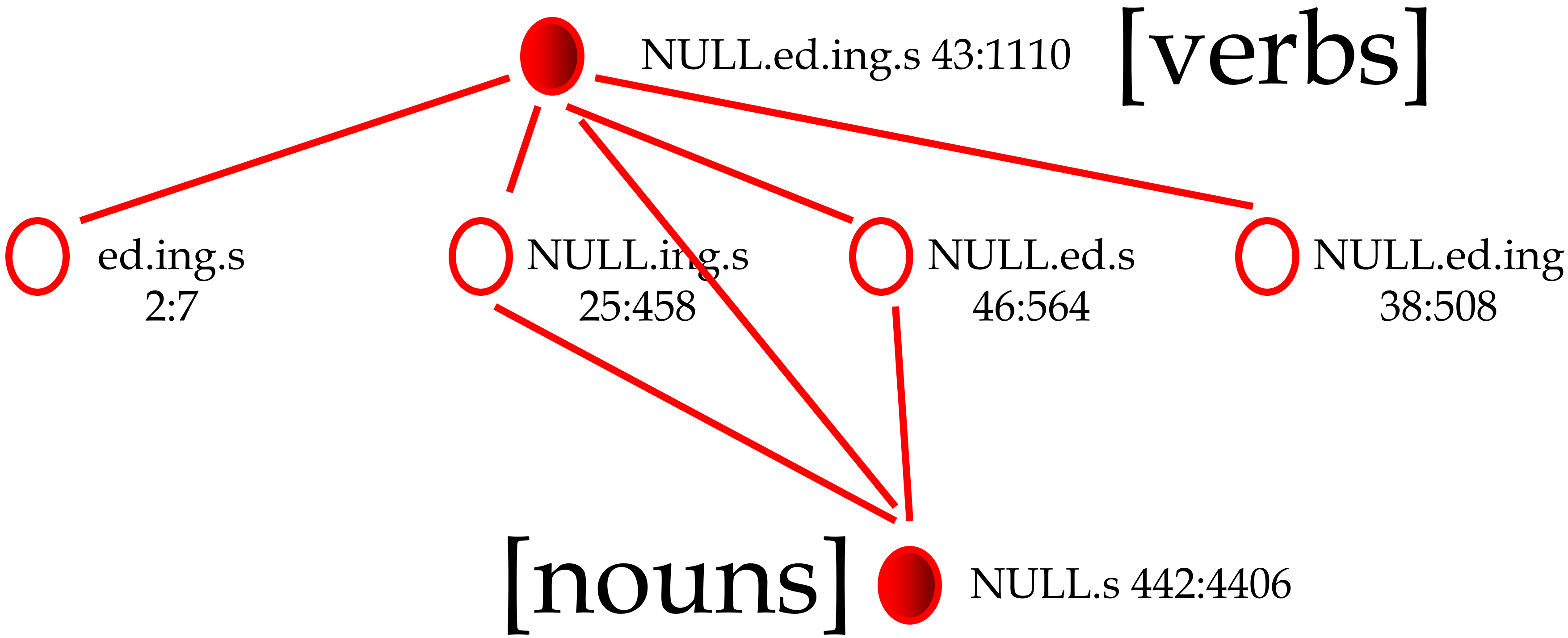
- What is human language? Find the most compact description of the Internet, where we assume that all data is labeled by the language it came from. Then: some *part* of the minimal description of that data is an answer to the question: what is human language.

What is linguistics?

- Linguistics is the application of algorithmic complexity analysis to language data.
- It is not necessary to specify a class of models in advance.
- If a linguist chooses to explore a specific class of models, that is an existential *bet* that this class of models is the best.
- But there is no guarantee.

- We have given you a small picture of the larger task of unsupervised learning of natural language structure using description length minimization.

The end



Generalization
consists of eliminating nodes,
and push their stems upward