A Shifting Perspective on Stem Space for French Verbs

Olivier Bonami & Gilles Boyé
Université Paris-La Sorbonne, LLF
Université de Bordeaux, CLLE
MorPa
Defining stem spaces

A Network of relations
Motivations

• In a Word and Paradigm framework, the lexeme is the unit binding together the various inflectional forms.

• In this context, two interesting conceptions:
  • From a psycholinguistic standpoint, can be seen as a network of relations linking every form to all the others in the same paradigm
  • From a linguistic standpoint, can be seen as a network linking all the forms of the paradigm to the principal parts.

• While the maximally connected network is easy to define, we show that it is not linguistically appropriate.
  • An appropriate network should constrain the distribution of irregularity to the observed patterns.
  • It should also accommodate different patterns of lexical storage for existing lexemes
Forms and paradigms

- A paradigm is a set of related forms
A paradigm is a set of related forms.
A paradigm is a set of related forms
Points of irregularity

- To be an irregular verb is to necessitate more than one principal part
- regular forms are connected

A Shifting Perspective on the Stem Space for French Verbs – 5
Points of irregularity

- To be an irregular verb is to necessitate more than one principal part
  - regular forms are connected
  - suppletive forms are isolated

A Shifting Perspective on the Stem Space for French Verbs – 5
Blocks of forms

- For future and conditional, the 12 forms are inter-predictable for all verbs:
  - 66 symmetric relations link all the forms together
Blocks of forms

- For future and conditional, the 12 forms are inter-predictable for all verbs:
  - 66 symmetric relations link all the forms together
  - 11 relations are sufficient for a linguistic description
Blocks of forms

- For future and conditional, the 12 forms are inter-predictable for all verbs:
  - 66 symmetric relations link all the forms together
  - 11 relations are sufficient for a linguistic description
    - 239,500,800 possible exhaustive paths
    - all paths are linguistically equivalent
  - the 12 forms constitute a inter-predictable block of inflectional morphology: a block of forms
• Between blocks, forms are sometimes inter-predictable
  • we could establish relations between all the forms between blocks
  • but one relation is sufficient
Links between blocks of forms

- Between blocks, forms are sometimes inter-predictable
- we could establish relations between all the forms between blocks
- but one relation is sufficient
Links between blocks of forms

- Between blocks, forms are sometimes inter-predictable
- we could establish relations between all the forms between blocks
- but one relation is sufficient
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface.
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface.
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface.
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface.
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface
Stem space

- The stem space captures the blocks of forms.
  - The forms in a block are all linked to the same stem.
- The stem graph captures the distribution of semi-regularity.
  - The stem relations constitute an acyclic undirected connected graph (i.e. a tree)
  - The paradigm is minimally connected
- Heuristics for constructing the stem graph:
  - Stem slots which hold related values when all others are different are linked directly.
  - If all the stem slots are related but one, that stem slot has only one link (tree leaf)
Modelling stem spaces
Two alternatives
Constructive vs. abstractive approaches

- Blevins (2006): distinction between constructive and abstractive approaches to morphology.
- In a constructive approach, the basic units are roots (or stems); rules of inflectional morphology specify how exponents are added/applied to roots/stems to specify words.
- In an abstractive approach, the basic units are words; rules of inflectional morphology specify how the word filling some cell of the paradigm can be determined on the basis of the content of other cells.
- Blevins’s choice of words is somewhat misleading: abstracting away stems (or roots) plays no role in the type of analysis he proposes.
- We suggest that stems can be useful within an abstractive approach.
A semi-constructive approach to stem spaces

- Bonami & Boyé (2006, 2007) propose an HPSG approach to French conjugation which is a hybrid between a constructive and abstractive approach:

- Stem spaces are complex representations within lexical entries, structured by default morphophonological relations.

- Word forms are deduced from the stem space using a paradigm function (Stump, 2001)

\[
\begin{array}{c|c|c}
\text{v-lexeme} & \text{HEAD} & \text{verb} \\
& \begin{bmatrix}
\text{STEMS} \\
\text{SLOT1} \\
\text{SLOT2} \\
\text{SLOT3} \\
\ldots
\end{bmatrix}
& \begin{bmatrix}
\text{v-stem-space} \\
\text{phon} \\
\text{phon} \\
\text{phon} \\
\end{bmatrix}
\end{array}
\]

\[
\begin{array}{c|c}
\text{boire:} & \text{STEMS} \\
& \begin{bmatrix}
\text{STEMS} \\
\text{SLOT1} \\
\text{SLOT2} \\
\text{SLOT3} \\
\end{bmatrix}
& \begin{bmatrix}
\text{v-stem-space} \\
\text{byv} \\
\text{bwav} \\
\text{bwa}
\end{bmatrix}
\end{array}
\]
Filling the stem space

(8) \( verb-lexeme \rightarrow [STEMS \mid regular] \)

(7) 

\[
\begin{array}{c}
\text{v-stem-space} \\
\begin{bmatrix}
s1-like-s2 \\
\text{SLOT1} \\
\text{SLOT2}
\end{bmatrix}
\end{array}
\]

\[
\begin{array}{c}
\text{fully-irreg} \\
\begin{bmatrix}
s3-like-s2 \\
\text{SLOT2} \\
\text{SLOT3}
\end{bmatrix}
\end{array}
\]

\[
\begin{array}{c}
\text{irreg-s3} \\
\begin{bmatrix}
\text{valoir} \\
\text{SLOT1} \text{val} \\
\text{SLOT2} \text{val} \\
\text{SLOT3} \text{vo}
\end{bmatrix}
\end{array}
\]

\[
\begin{array}{c}
\text{irreg-s1} \\
\begin{bmatrix}
\text{mournir} \\
\text{SLOT1} \text{mur} \\
\text{SLOT2} \text{mœr} \\
\text{SLOT3} \text{mœr}
\end{bmatrix}
\end{array}
\]

\[
\begin{array}{c}
\text{regular} \\
\begin{bmatrix}
\text{laver} \\
\text{SLOT1 lav} \\
\text{SLOT2 lav} \\
\text{SLOT3 lav}
\end{bmatrix}
\end{array}
\]

\[
\begin{array}{c}
\text{boire} \\
\begin{bmatrix}
\text{SLOT1 byv} \\
\text{SLOT2 bwav} \\
\text{SLOT3 bwa}
\end{bmatrix}
\end{array}
\]

A Shifting Perspective on the Stem Space for French Verbs – 13
The PFM component

Block 3

a. $X_V, \sigma : \{\text{RELTYPE } an\} \rightarrow X \oplus \varepsilon$

b. $X_V, \sigma : \{\text{RELTYPE } an\, \text{PER } 1, \text{NB } pl\} \rightarrow X \oplus j$

c. $X_V, \sigma : \{\text{MODE } subj, \text{PER } 1, \text{NB } pl\} \rightarrow X \oplus j$

d. $X_V, \sigma : \{\text{PER } 2, \text{NB } pl\} \rightarrow \langle X, \sigma / \{\text{PER } 1\} \rangle : 3$
A fully abstractive alternative

- Come back to the view of inflection as an oriented graph
- Arcs within the graph can be modelled by standard lexical rules
- Stems are just nodes with no morphosyntactic description
- See e.g. the indicative present:
A fully abstractive alternative

- Determining the form filling a slot amounts to circulating the graph until one finds a lexical entry.
- To inflect correctly *laver* in the 1pl, the speaker can rely on the knowledge of any other form or stem. Suppose he knows only the 1sg:

\[
\begin{align*}
&\tau = \text{lav} \\
&Z = \tau = \text{lav} \\
&\gamma = Z = \text{lav} \\
&x = \gamma \oplus o = \text{lav}o
\end{align*}
\]
A fully abstractive alternative

- Determining the form filling a slot amounts to circulating the graph until one finds a lexical entry.
- To inflect correctly *savoir* in the 1pl, the speaker needs to know more specific information—e.g. stem 2.
HPSG implementation: constraints on types

- Lexemes do not exist as data structures. Rather, a collection of lexical objects share a *lexemic index* (à la Spencer)

- Two types of lexical objects:
  - Inflectional relations specify how some form of a lexeme depends on another form of the same lexeme
  - Lexical entries are lexical objects that do not depend on any other lexical object

- Lexical entries win over inflectional relations

\[
\text{lex-obj} \rightarrow \left[ \begin{array} { c } { \text{LXM} } \\ { \text{lxm-ind} } \end{array} \right]
\]

\[
\text{infl-rln} \rightarrow \left[ \begin{array} { c } { \text{LXM} } \\ { \text{\[\]}} \end{array} \right]
\]

\[
\text{lex-entry} \rightarrow \left[ \begin{array} { c } { \text{DTRS} } \\ { \text{\{} } \end{array} \right]
\]

\[
\text{lex-obj} \rightarrow (\text{lex-entry} \lor (\neg\text{lex-entry} \land \text{infl-rln}))
\]
HPSG implementation: the type hierarchy

- More technically:

```
syntagmatic-obj

lex-obj

LEX-STATUS  ABSTRACTION  sign

lex-entry  infl-rln  stem  word  phrase

stm-le  stm-ir  wd-le  wd-ir

syntagmatic-obj → [PHON  phon]  sign → [SYNSEM  synsem]
```
HPSG implementation: the network

Concentrating again on the present indicative:

\[
\text{infl-rln} \rightarrow
\begin{align*}
\text{PHON} & \quad \llbracket \text{PHON} \rrbracket \\
\text{SYNSEM} & \quad \llbracket \text{sg} \rrbracket \\
\text{DTRS} & \quad \llbracket \text{PHON} \rrbracket \\
\text{SS} & \quad \llbracket \text{3pl} \rrbracket \\
\text{DTRS} & \quad \llbracket \text{PHON} \rrbracket \\
\text{STM-IND} & \quad \llbracket 1 \rrbracket \\
\text{DTRS} & \quad \llbracket \text{PHON} \rrbracket \\
\text{SS} & \quad \llbracket [1pl] \rrbracket \\
\text{DTRS} & \quad \llbracket \text{PHON} \rrbracket \\
\text{STM-IND} & \quad \llbracket 3 \rrbracket \\
\text{DTRS} & \quad \llbracket \text{PHON} \rrbracket \\
\text{SS} & \quad \llbracket [2pl] \rrbracket \\
\text{DTRS} & \quad \llbracket \text{PHON} \rrbracket \\
\text{STM-IND} & \quad \llbracket 2 \rrbracket \\
\text{DTRS} & \quad \llbracket \text{PHON} \rrbracket \\
\text{SS} & \quad \llbracket [3pl] \rrbracket \\
\text{DTRS} & \quad \llbracket \text{PHON} \rrbracket \\
\text{STM-IND} & \quad \llbracket 1 \rrbracket \\
\end{align*}
\]
HPSG implementation: the lexical entries

• Minimal specifications for some lexemes:

\[
\begin{bmatrix}
\text{lex-entry} \\
\text{LXM laver}
\end{bmatrix} \rightarrow \begin{bmatrix}
\text{PHON lav} \\
\text{STM-IND 1}
\end{bmatrix}
\]
HPSG implementation: the lexical entries

- Minimal specifications for some lexemes:

\[
\begin{align*}
\text{lex-entry} & \rightarrow \begin{bmatrix}
\text{PHON} & \text{mur} \\
\text{LXM} & \text{mourir} \\
\text{STM-IND} & 1
\end{bmatrix} \lor \begin{bmatrix}
\text{PHON} & \text{mœr} \\
\text{STM-IND} & 2
\end{bmatrix}
\end{align*}
\]
HPSG implementation: the lexical entries

- Minimal specifications for some lexemes:

\[
\begin{align*}
\text{[lex-entry] :} & \quad \text{[PHON \ byv]} \lor \text{[PHON \ bwav]} \\
\text{LXM boire :} & \quad \text{[STM-IND 1]} \lor \text{[STM-IND 2]} \\
\end{align*}
\]

A Shifting Perspective on the Stem Space for French Verbs – 21
HPSG implementation: the lexical entries

- Minimal specifications for some lexemes:

\[
\begin{align*}
\text{lex-entry} & \rightarrow \left[ \begin{array}{c}
\text{PHON} \ fœz \\
\text{STM-IND} \ 1
\end{array} \right] \lor \left[ \begin{array}{c}
\text{PHON} \ fô \\
\text{STM-IND} \ 2
\end{array} \right] \lor \left[ \begin{array}{c}
\text{PHON} \ fε \\
\text{STM-IND} \ 3
\end{array} \right] \lor \left[ \begin{array}{c}
\text{SS} \\
[2\text{pl}]
\end{array} \right]
\end{align*}
\]

A Shifting Perspective on the Stem Space for French Verbs – 21
HPSG implementation: the lexical entries

- Minimal specifications for some lexemes:

\[
\text{LXM} \; \text{être} \rightarrow \left[ \text{lex-entry} \right] \left[ \begin{array}{c}
\text{PHON} \; \varepsilon \text{t} \\
\text{STM-IND} \; 1
\end{array} \right] \lor \left[ \begin{array}{c}
\text{PHON} \; s\ddot{o} \\
\text{STM-IND} \; 2
\end{array} \right] \lor \left[ \begin{array}{c}
\text{PHON} \; \varepsilon \\
\text{STM-IND} \; 3 \\
\text{SS} \; [1sg]
\end{array} \right] \lor \left[ \begin{array}{c}
\text{PHON} \; s\ddot{u}i \\
\text{SS} \; [1pl]
\end{array} \right]
\]

[prst, indic, 2pl]  
\[
\varepsilon \text{t}
\]

[prst, indic, 1pl]  
\[
s\ddot{u}i
\]

[prst, indic, 3pl]  
\[
\text{ssc}
\]

[prst, indic, 2sg]  
\[
\varepsilon
\]

[prst, indic, 3sg]  
\[
\text{ssc}
\]

[prst, indic, 1sg]  
\[
s\ddot{u}i
\]
The abstractive approach: evaluation

- Shows how an abstractive approach can accommodate stem spaces.

- From a technical, HPSG point of view:
  - Captures the relevant generalizations without appealing to disputed apparatus (online type construction, defaults).

- Psycholinguistically realistic:
  - The amount of stored lexical knowledge for two lexemes in the same class need not be the same.
  - Only principal parts are needed, but more can be stored.

- Linguistically minimal:
  - Only with very few lexemes is it necessary to store redundant information.
  - These correspond roughly to the *complex dependency relations* left aside in Bonami & Boyé 2002.
The abstractive approach: evaluation

• Exponence issue:
  • We have no principled decomposition of exponence à la PFM. All endings are atomic.
  • But in fact we do not loose much:
    • the PFM analysis in Bonami & Boyé (2007a) uses 19 rules of exponence + 13 rules of stem selection to generate 48 forms.
    • here we use 48 rules which deal simultaneously with exponence and stem selection

• Redundancy issue:
  • We need to store converse inflectional relations systematically, which seems redundant.
  • as it turns out, the relevant relations are often not really converses… see section 3
Extensions
Problem 1: converse opacities

- Assume that regular conjugation patterns in French correspond to the 1st and 2nd traditional groups (Bonami, Boyé, Giraudo & Vog, 2008)
- Then there are at least 6 distinct regular patterns

<table>
<thead>
<tr>
<th>Verb</th>
<th>PRST 1SG</th>
<th>PRST 2SG</th>
<th>PRST 3SG</th>
<th>PRST 1PL</th>
<th>PRST 2PL</th>
<th>PRST 3PL</th>
<th>INF</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPUYER</td>
<td>apųi</td>
<td>apųi</td>
<td>apųi</td>
<td>apųij</td>
<td>apųij</td>
<td>apųi</td>
<td>apųije</td>
</tr>
<tr>
<td>AIGUILLER</td>
<td>ęgųij</td>
<td>ęgųij</td>
<td>ęgųij</td>
<td>ęgųij</td>
<td>ęgųij</td>
<td>ęgųij</td>
<td>ęgųije</td>
</tr>
<tr>
<td>ABOYER</td>
<td>abwa</td>
<td>abwa</td>
<td>abwa</td>
<td>abwaj</td>
<td>abwaj</td>
<td>abwa</td>
<td>abwaje</td>
</tr>
<tr>
<td>CRÉER</td>
<td>kre</td>
<td>kre</td>
<td>kre</td>
<td>kre</td>
<td>kre</td>
<td>kre</td>
<td>kree</td>
</tr>
<tr>
<td>TAPISSE</td>
<td>tapis</td>
<td>tapis</td>
<td>tapis</td>
<td>tapis</td>
<td>tapis</td>
<td>tapis</td>
<td>tapise</td>
</tr>
<tr>
<td>TAPIR</td>
<td>tapi</td>
<td>tapi</td>
<td>tapi</td>
<td>tapis</td>
<td>tapis</td>
<td>tapis</td>
<td>tapir</td>
</tr>
</tbody>
</table>
Problem 1: converse opacities

- For some verbs, prediction from stem 1 does not work
Problem 1: converse opacities

- Neither does stem 2
Problem 1: converse opacities

- Stem 3 does not work either
Problem 1: converse opacities

- Same reasoning holds for all other stems or forms, e.g. the infinitive

TRIER

trij
[STEM-IND 1]

tri
[STEM-IND 2]

tri
[STEM-IND 3]

TRILLER

trij
[STEM-IND 1]

trije
[STEM-IND 9]

trij
[STEM-IND 3]
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
Solution 1: more connected, less predictive

• Change connections to partial functions
• Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached.
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
- Note that:
  - There is still only one principal part for each regular French verb
  - Storing redundant information in the lexicon still works
  - Stating separately the converse connections between two slots is a good idea after all
### Problem 2: complex dependencies

- There are cases where a stem (resp. a form) can only be predicted on the basis of two other stems (resp. stems, forms).
- A simple example: Masculine singular liaison forms of French adjectives (Bonami & Boye, 2005)

<table>
<thead>
<tr>
<th></th>
<th>MAS.SG</th>
<th>FEM.SG</th>
<th>MSLF</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOLI</td>
<td>ʒoli</td>
<td>ʒoli</td>
<td>ʒoli</td>
</tr>
<tr>
<td>NET</td>
<td>nɛt</td>
<td>nɛt</td>
<td>nɛt</td>
</tr>
<tr>
<td>PETIT</td>
<td>pətɪ</td>
<td>pətɪt</td>
<td>pətɪt</td>
</tr>
<tr>
<td>COURT</td>
<td>kur</td>
<td>kurt</td>
<td>kurt</td>
</tr>
<tr>
<td>NOUVEAU</td>
<td>nuvo</td>
<td>nuvɛl</td>
<td>tapi</td>
</tr>
</tbody>
</table>

⇒ The MSLF is identical to the FEM.SG if the MAS.SG is vocalic; else it is identical to the MAS.SG.

- Other examples: languages with multiple principal parts for regular lexemes.
Problem 2: complex dependencies

- There are cases where a stem (resp. a form) can only be predicted on the basis of two other stems (resp. stems, forms)
- In French conjugation: past participles (Bonami & Boye, 2006b)
  - For regular verbs participle formation is fully predictable from the infinitive stem: i → i, e → e
  - For irregulars, many attested patterns. But speakers have a strong preference for participles in y (see e.g. Kilani-Schoch & Dressler, 2005)

<table>
<thead>
<tr>
<th></th>
<th>INF</th>
<th>TRUE PP</th>
<th>COMMON ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAVER</td>
<td>lave</td>
<td>lave</td>
<td>—</td>
</tr>
<tr>
<td>FINIR</td>
<td>fini</td>
<td>fini</td>
<td>—</td>
</tr>
<tr>
<td>BOIRE</td>
<td>bwa</td>
<td>by</td>
<td>—</td>
</tr>
<tr>
<td>MORDRE</td>
<td>mɔrd</td>
<td>mɔrdy</td>
<td>—</td>
</tr>
<tr>
<td>PRENDRE</td>
<td>prän≤d</td>
<td>priz</td>
<td>prän≤dy</td>
</tr>
<tr>
<td>MOURIR</td>
<td>muri</td>
<td>mɔrt</td>
<td>mury</td>
</tr>
<tr>
<td>PEINDRE</td>
<td>pẽd</td>
<td>pẽt</td>
<td>pẽdy</td>
</tr>
</tbody>
</table>
### Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:

<table>
<thead>
<tr>
<th></th>
<th>INF</th>
<th>SIMPLE PAST</th>
<th>TRUE PP</th>
<th>COMMON ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAVER</td>
<td>lave</td>
<td>lava</td>
<td>lave</td>
<td>—</td>
</tr>
<tr>
<td>FINIR</td>
<td>fini</td>
<td>fini</td>
<td>fini</td>
<td>—</td>
</tr>
<tr>
<td>BOIRE</td>
<td>bwa</td>
<td>by</td>
<td>by</td>
<td>—</td>
</tr>
<tr>
<td>MORDRE</td>
<td>mɔrd</td>
<td>mɔrdi</td>
<td>mɔrdy</td>
<td>—</td>
</tr>
<tr>
<td>PRENDRE</td>
<td>prãd</td>
<td>pri</td>
<td>priz</td>
<td>prãdy</td>
</tr>
<tr>
<td>MOURIR</td>
<td>muri</td>
<td>mury</td>
<td>mɔrt</td>
<td>mury</td>
</tr>
<tr>
<td>PEINDRE</td>
<td>pẽd</td>
<td>pẽnji</td>
<td>pẽt</td>
<td>pẽdy</td>
</tr>
</tbody>
</table>
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - in the default case, predict both stem 11 and stem 12 from stem 9
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
- in the default case, predict both stem 11 and stem 12 from stem 9
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
- in the default case, predict both stem 11 and stem 12 from stem 9
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - if stem 11 is not as predicted, then predict stem 12 from stem 11
  - if stem 12 is not as predicted, then store it
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - if stem 11 is not as predicted, then predict stem 12 from stem 11
  - if stem 12 is not as predicted, then store it
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - if stem 11 is not as predicted, then predict stem 12 from stem 11
  - if stem 12 is not as predicted, then store it

A Shifting Perspective on the Stem Space for French Verbs – 39
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
- Errors arise when the speaker forgets about an unpredictable stem
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
- Errors arise when the speaker forgets about an unpredictable stem
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
- Errors arise when the speaker forgets about an unpredictable stem
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
- Errors arise when the speaker forgets about an unpredictable stem
Solution 2: n-ary inflectional dependencies

- Up to now
  - The inflectional network is a graph connecting pairs of paradigm elements
  - The arcs in the graph are modelled by feature structure descriptions with a single DTRS element.

- To account for complex dependencies we generalize
  - From pairs of paradigm elements to tuples of paradigm elements
  - Thus, to a network that is not a graph (not defined by a binary relation)
  - Formally, complex dependencies are modelled by feature structure descriptions with more than one DTRS element
Conclusions
Summing up

• Starting from the view of inflection as a network of forms:
  • We argued that stem spaces are motivated in French to account for suppletion patterns
  • We showed that stem spaces are fully compatible with an abstractive view of inflection
  • We discussed the benefits of fine-tuning the connectedness of the network of stems and forms
• At first sight this goes against Blevins’s (2006) conclusions that:
  • Stems and roots are usually useless for the description of inflection patterns
  • In many cases starting from stems does not work, except through the postulation of nonobservables (class features, empty morphs, abstract phonology)
Summing up

• While we agree with the diagnosis, we think the illness comes from the search for minimal meaningful units, rather than abstract inflectional objects.

• Take French infinitives and present singulars:

<table>
<thead>
<tr>
<th></th>
<th>INF</th>
<th>PRST.SG</th>
<th>STEM 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>LÄVER</td>
<td>lave</td>
<td>lav</td>
<td>lave</td>
</tr>
<tr>
<td>FINIR</td>
<td>finir</td>
<td>fini</td>
<td>fini</td>
</tr>
<tr>
<td>BOIRE</td>
<td>bwar</td>
<td>bwa</td>
<td>bwa</td>
</tr>
<tr>
<td>MORDRE</td>
<td>mördr</td>
<td>mör</td>
<td>mörd</td>
</tr>
<tr>
<td>SORTIR</td>
<td>sortir</td>
<td>sort</td>
<td>sort</td>
</tr>
</tbody>
</table>

• Traditional approaches try to abstract away a minimal stem, and then need class features to account for infinitive endings

• In the current approach the choice of the stem is determined by suppletion patterns rather than minimality conditions

• Thus the stem is informative enough to serve as a principal part
An empirical test

• Thus the stem space is fundamentally a hypothesis on the relationship between stem suppletion and exponence

• What would be problematic would be a language where:
  • suppletion patterns motivate a stem space; but
  • prediction relations can not be seen on the stems

<table>
<thead>
<tr>
<th></th>
<th>MAS.SG</th>
<th>MAS.PL</th>
<th>FEM.SG</th>
<th>FEM.PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bak</td>
<td>bata</td>
<td>batu</td>
<td>baʃ</td>
<td></td>
</tr>
<tr>
<td>lup</td>
<td>luti</td>
<td>lutu</td>
<td>luʃ</td>
<td></td>
</tr>
<tr>
<td>mik</td>
<td>mita</td>
<td>mite</td>
<td>mis</td>
<td></td>
</tr>
<tr>
<td>sup</td>
<td>suti</td>
<td>sute</td>
<td>sus</td>
<td></td>
</tr>
<tr>
<td>sik</td>
<td>keta</td>
<td>kete</td>
<td>sis</td>
<td></td>
</tr>
<tr>
<td>uk</td>
<td>ama</td>
<td>amu</td>
<td>uʃ</td>
<td></td>
</tr>
<tr>
<td>pap</td>
<td>tisi</td>
<td>tisu</td>
<td>paʃ</td>
<td></td>
</tr>
</tbody>
</table>
An empirical test

• Thus the stem space is fundamentally a hypothesis on the relationship between stem suppletion and exponence

• What would be problematic would be a language where:
  • suppletion patterns motivate a stem space; but
  • prediction relations can not be seen on the stems

<table>
<thead>
<tr>
<th>MAS.SG</th>
<th>MAS.PL</th>
<th>FEM.SG</th>
<th>FEM.PL</th>
<th>STEM 1</th>
<th>STEM 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>bak</td>
<td>bata</td>
<td>batu</td>
<td>baf</td>
<td>ba</td>
<td>bat</td>
</tr>
<tr>
<td>lup</td>
<td>luti</td>
<td>lutu</td>
<td>luʃ</td>
<td>lu</td>
<td>lut</td>
</tr>
<tr>
<td>mik</td>
<td>mita</td>
<td>mite</td>
<td>mis</td>
<td>mi</td>
<td>mit</td>
</tr>
<tr>
<td>sup</td>
<td>suti</td>
<td>sute</td>
<td>sus</td>
<td>su</td>
<td>sut</td>
</tr>
<tr>
<td>sik</td>
<td>keta</td>
<td>kete</td>
<td>sis</td>
<td>si</td>
<td>ket</td>
</tr>
<tr>
<td>uk</td>
<td>ama</td>
<td>amu</td>
<td>uʃ</td>
<td>u</td>
<td>am</td>
</tr>
<tr>
<td>pap</td>
<td>tisi</td>
<td>tisu</td>
<td>paʃ</td>
<td>pa</td>
<td>tis</td>
</tr>
</tbody>
</table>
An empirical test

- Thus the stem space is fundamentally a hypothesis on the relationship between stem suppletion and exponence.

- What would be problematic would be a language where:
  - suppletion patterns motivate a stem space; but
  - prediction relations can not be seen on the stems.