Automatic Segmentation of Transitive Paradigms

Sebastian Bank

University of Leipzig

Workshop on the Fine Structure of Grammatical Relations
December 10, 2010

Outline

Automatic Segmentation of Transitive Paradigms

The Problem

How to analyze (complex) inflectional systems?

(1) *Ainu*

A→P	1sg	1pl	2sg	2pl	3sg	3pl	indef
1sg	-	-	eci	eci	ku	ku	kui
1pl	-	-	eci	eci	ci	ci	ai
2sg	en	un	-	-	e	e	ei
2pl	ecien	eciun	-	-	eci	eci	ecii
3sg	en	un	e	eci	-	-	i
3pl	en	un	e	eci	-	-	i
indef	aen	aun	ae	aeci	a	a	ai

(Tamura 2000)

Goals

Possible benefits from algorithmic paradigm learning and segmentation

- Assist concrete analyses
  - necessary condition for the occurence of a marker
  - candidates for the sufficient condition
  - candidates for blocking rules, impoverishment rules, homonyms
  - alternative segmentations
- Capture/compare cross-linguistic tendencies and patterns
  - prevailing marker types, deviance
  - neutralizations, classify syncretism types
  - 'biased' segmentation
- Implications for language change, variation and acquisition
  - 'critical paradigm cells'
  - recutting, reanalysis
- Improve understanding of the theoretical basis
  - possible syncretism types, unattested syncretisms
  - properties of the necessary formalisms

Learning Markers

Finding the invariant meaning for a form

(2) *Determining the invariant meaning of a syncretic marker by intersection (cf. Pertsova 2007)*

a.

	sg	**pl**
1	I	we
2	you	you

b.

form	occurrences	invariance
I	[1,sg]	[1,sg]
we	[1,pl]	[1,pl]
you	[2,sg],[2,pl]	[2]

(3) a. *you* → [2] b. *you* ← [2]

(4) a. *I* ↔ [1,sg] b. *we* ↔ [1,pl] c. *you* ↔ [2]

(5) *Different marker distributions and their invariant meaning by intersection*

a. (i)

	sg	**pl**
1	A	D
2	B	D
3	C	C

(ii)

	sg	**pl**
1	E	F
2	E	G
3	G	G

b.

form	occurrences	invariance	relation
A	[+1,+sg]	[+1,+sg]	=
B	[+2,+sg]	[+2,+sg]	=
C	[+3,+sg],[+3,+pl]	[+3]	↔
D	[+1,+pl],[+2,+pl]	[−3,+pl]	↔
E	[+1,+sg],[+2,+sg]	[−3,+sg]	↔
F	[+1,+pl]	[+1,+pl]	=
G	[+2,+pl],[+3,+sg],[+3,+pl]	[−1]	→

(6) a. G → [−1] b. E ∨ G ← [−1] c. G ↞ [−1]

Learning Transitive Agreement: Portmanteau and Object Marker

(7) a.

S	
1	A
2	B
3	C

b.

A→P	1	2	3
1	A	B	EF
2	AB	B	BDF
3	AC	BCD	CF

(8)

form	occurrences	invariance
E	[A,+1][P,+3]	[A,+1][P,+3]

(9)

form	occurrences	invariance
F	[A,+1][P,+3], [A,+2][P,+3], [A,+3][P,+3]	[A][P,+3]

Learning Transitive Agreement: Nominative and Absolutive

(10) a.

S	
1	A
2	B
3	C

b.

A→P	1	2	3
1	A	B	EF
2	AB	B	BDF
3	AC	BCD	CF

(11)

form	occurrences	invariance
A	[S,SA,SP,+1], [A,SA,+1][P,SP,+1], [A,SA,+2][P,SP,+1], [A,SA,+3][P,SP,+1]	[SA][SP,+1]

(12)

form	occurrences	invariance
C	[S,SA,SP,+3], [A,SA,+3][P,SP,+1], [A,SA,+3][P,SP,+2], [A,SA,+3][P,SP,+3]	[SA,+3][SP]

Learning Transitive Agreement: Neutral and Mirror-Portmanteau

(13)

a.

S	
1	A
2	B
3	C

b.

A→P	1	2	3
1	A	B	EF
2	AB	B	BDF
3	AC	BCD	CF

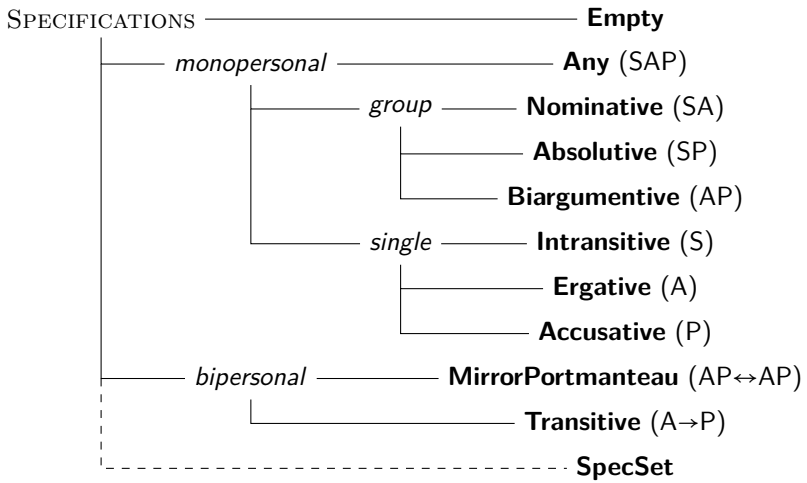
(14)

form	occurrences	invariance
B	[S,SA,SP,+2], [A,SA,+1][P,SP,+2], [A,SA,+2][P,SP,+1], [A,SA,+2][P,SP,+2] [A,SA,+2][P,SP,+3], [A,SA,+3][P,SP,+2]	[+2][]

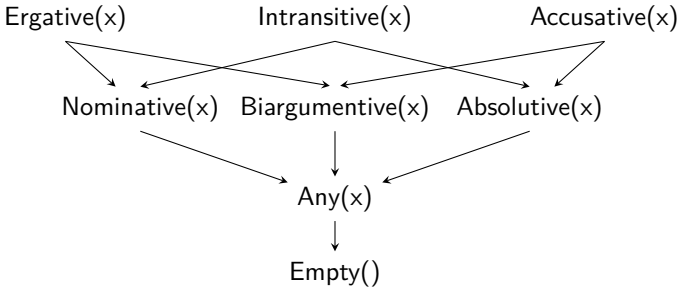
(15)

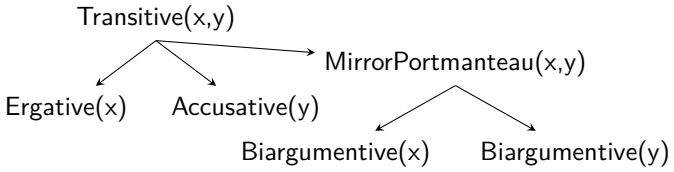
form	occurrences	invariance
D	[A,+3][P,+2], [A,+2][P,+3]	[+3][+2]

Specifications for Transitive Agreement



The Combinatorics of Specifications





Practical Application: Thulung Non-past (Lahaussois 2002)

1s	ŋu
1de	tsuku
1pe	ku
1di	tsi
1pi	di
2s	na
2d	tsi
2p	ni
3s	
3d	tsi
3p	mi

	1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s	-	-	-	-	-	ni	ni-tsi	ni-ni	u	u-tsi	u-mi
1de	-	-	-	-	-	tsuku	tsuku	tsuku	tsuku	tsuku	tsuku
1pe	-	-	-	-	-	ku	ku	ku	ku	ku	ku
1di	-	-	-	-	-	-	-	-	tsi	tsi	tsi
1pi	-	-	-	-	-	-	-	-	i	i-tsi	i
2s	ŋi	tsi-ki	ki	-	-	-	-	-	na	na	na
2d	ŋi-tsi	tsi-ki	tsi-ki	-	-	-	-	-	tsi	tsi	tsi
2p	ŋi-ni	ki-ni	ki-ni	-	-	-	-	-	ni	ni	ni
3s	ŋi	tsi-ki	ki	tsi-ki	sa	na	na	ni-mi	y	y-tsi	y-mi
3d	ŋi-tsi	tsi-ki	ki-ni	sa	sa-mi	na-tsi	na-tsi	ni-tsi	y-tsi	y-tsi	y-tsi
3p	ŋi-mi	tsi-ki	ki-mi	sa-mi	sa-mi	na-mi	na-tsi-mi	ni-mi	mi	mi	mi

Nominative(+1,−2,+pl)

		1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s		1s										
1de		1de										
1pe	ku	1pe	?	?	?	?	?	ku	ku	ku	ku	ku	ku
1di		1di										
1pi		1pi										
2s		2s										
2d		2d										
2p		2p										
3s		3s										
3d		3d										
3p		3p										

```
ku  <->  S[+1,-2,-3,-sg,-du,+pl],
         [+1,-2,-3,-sg,-du,+pl]A->P[-1,+2,-3,+sg,-du,-pl],
         [+1,-2,-3,-sg,-du,+pl]A->P[-1,+2,-3,-sg,+du,-pl],
         [+1,-2,-3,-sg,-du,+pl]A->P[-1,+2,-3,-sg,-du,+pl],
         [+1,-2,-3,-sg,-du,+pl]A->P[-1,-2,+3,+sg,-du,-pl],
         [+1,-2,-3,-sg,-du,+pl]A->P[-1,-2,+3,-sg,+du,-pl],
         [+1,-2,-3,-sg,-du,+pl]A->P[-1,-2,+3,-sg,-du,+pl]
       -> SA[+1,-2,-3,-sg,-du,+pl]
      <-> yes (100%)
```


Ergative(+1,+sg)

	1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s	?										
1de	?										
1pe	?										
1di	?										
1pi	?										
2s	ŋi										
2d	ŋi										
2p	ŋi										
3s	ŋi										
3d	ŋi										
3p	ŋi										

```
?i  <->  [-1,+2,-3,+sg,-du,-pl]A->P[+1,-2,-3,+sg,-du,-pl],
         [-1,+2,-3,-sg,+du,-pl]A->P[+1,-2,-3,+sg,-du,-pl],
         [-1,+2,-3,-sg,-du,+pl]A->P[+1,-2,-3,+sg,-du,-pl],
         [-1,-2,+3,+sg,-du,-pl]A->P[+1,-2,-3,+sg,-du,-pl],
         [-1,-2,+3,-sg,+du,-pl]A->P[+1,-2,-3,+sg,-du,-pl],
         [-1,-2,+3,-sg,-du,+pl]A->P[+1,-2,-3,+sg,-du,-pl]
      ->  P[+1,-2,-3,+sg,-du,-pl]
     <->  yes (100%)
```


		1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s		1s										
1de		1de										
1pe		1pe										
1di		1di										
1pi		1pi										
2s	na	2s						?	?	?	na	na	na
2d	+	2d						?	?	?	+	+	+
2p		2p						?	?				
3s		3s						na	na				
3d		3d						na	na				
3p		3p						na	na				

```
na  <->  2s 2s->3s 2s->3d 2s->3p 3s->2s 3s->2d 3d->2s 3d->2d 3p->2s
         3p->2d (10)
      ->  SA[-1] & SAP[-1,+2,-3,-pl] & SP[-1]
      ->  2s 2d |2s->2s| |2s->2d| |2s->2p| 2s->3s 2s->3d 2s->3p
          |2d->2s| |2d->2d| |2d->2p| 2d->3s 2d->3d 2d->3p |2p->2s|
          |2p->2d| 3s->2s 3s->2d 3d->2s 3d->2d 3p->2s 3p->2d (22)
      ?   |2s->2s| |2s->2d| |2s->2p| |2d->2s| |2d->2d| |2d->2p|
          |2p->2s| |2p->2d| (8)
      +   2d 2d->3s 2d->3d 2d->3p (4)
     <->  no (71.43%) (10 of 14)
```

Ergative(+1,–sg)

	1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s		?	?	?	?						
1de		?	?	?	?						
1pe		?	?	?	?						
1di		?	?	?	?						
1pi		?	?	?	?						
2s		ki	ki	?	?						
2d		ki	ki	?	?						
2p		ki	ki	?	?						
3s		ki	ki	ki	+						
3d		ki	ki	+	+						
3p		ki	ki	+	+						

```
ki  <->  2s->1de 2s->1pe 2d->1de 2d->1pe 2p->1de 2p->1pe 3s->1de
         3s->1pe 3s->1di 3d->1de 3d->1pe 3p->1de 3p->1pe (13)
     ->  P[+1,-3,-sg]
     +   3s->1pi 3d->1di 3d->1pi 3p->1di 3p->1pi (5)
    <->  no (72.22%) (13 of 18)
```


Division of Labor

Framework

- implement a module to represent different feature systems with their (in-)compatibilities and implications between features
- provide specification algebra: intersection, union, difference (drop invariances stemming from blind paradigm cells)
- implement subsets (power set) and supersets of specifications (generalization and specialization of markers)

Concrete Learners/Segmenters:

- rate/filter specifications (e.g. monopersonal > bipersonal, absolutive > nominative, etc.), translate if necessary
- provide (numeric) criteria for when to apply impoverishment, blocking, segmentation, and homonyms
- cope with affix-order/slots

Generalizations of a Specification

		1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s	1s	*										
1de	1de	*										
1pe	1pe	*										
1di	1di	*										
1pi	1pi	*										
2s	2s	*										
2d	2d	*										
2p	2p	*										
3s	3s	*										
3d	3d	*										
3p	3p	*										

P[-2,-3,-du,-pl]

Specialization of a Marker

	1sg	1pl	2sg	2pl	3sg	3pl	x
1sg			eci	eci			
1pl			eci	eci			
2sg					+	+	
2pl					eci	eci	
3sg			+	eci			
3pl			+	eci			
x							

[-x]A->P[-1,-x] & AP[-2,-x] & AP[-1,-3,-x]

Segmentation

Naive Approach: form based segmentation

```
while True:
    for form,cell in product(paradigm.morphemes(),paradigm.cells()):
        if cell.can_detach(form):
            cell.detach(form)
            break
    else:
        break
```

1. Collect the set of all forms M
2. For each form *f* from *M*, for each cell *C* in the paradigm, for each segment *s* from C:
   - If *f* is a proper substring of *s*,
     separarte it in *C* from the rest of *s*, and go to 1
3. Display the paradigm

na *natsi* *utsi*	*na* *na-tsi* *utsi*	*na* *na-tsi* *u-tsi*
M={*na*, *natsi*, *utsi*}	M={*na*, *tsi*, *utsi*}	M={*na*, *tsi*, *u*}

Segmentation: Result

Naive Approach: form based segmentation

```
    |
----+---------
1s  | ŋ-u
1de | t-s-u-k-u
1pe | k-u
1di | t-s-i
1pi | d-i
2s  | n-a
2d  | t-s-i
2p  | n-i
3s  |
3d  | t-s-i
3p  | m-i
```

```
    | 1s        | 1de       | 1pe       | 1di       | 1pi     | 2s        | 2d            | 2p        | 3s        | 3d        | 3p
----+-----------+-----------+-----------+-----------+---------+-----------+---------------+-----------+-----------+-----------+-----------
1s  | -         | -         | -         | -         | -       | n-i       | n-i-t-s-i     | n-i-n-i   | u         | u-t-s-i   | u-m-i
1de | -         | -         | -         | -         | -       | t-s-u-k-u | t-s-u-k-u     | t-s-u-k-u | t-s-u-k-u | t-s-u-k-u | t-s-u-k-u
1pe | -         | -         | -         | -         | -       | k-u       | k-u           | k-u       | k-u       | k-u       | k-u
1di | -         | -         | -         | -         | -       | -         | -             | -         | t-s-i     | t-s-i     | t-s-i
1pi | -         | -         | -         | -         | -       | -         | -             | -         | i         | i-t-s-i   | i
2s  | ŋ-i       | t-s-i-k-i | k-i       | -         | -       | -         | -             | -         | n-a       | n-a       | n-a
2d  | ŋ-i-t-s-i | t-s-i-k-i | t-s-i-k-i | -         | -       | -         | -             | -         | t-s-i     | t-s-i     | t-s-i
2p  | ŋ-i-n-i   | k-i-n-i   | k-i-n-i   | -         | -       | -         | -             | -         | n-i       | n-i       | n-i
3s  | ŋ-i       | t-s-i-k-i | k-i       | t-s-i-k-i | s-a     | n-a       | n-a           | n-i-m-i   | y         | y-t-s-i   | y-m-i
3d  | ŋ-i-t-s-i | t-s-i-k-i | k-i-n-i   | s-a       | s-a-m-i | n-a-t-s-i | n-a-t-s-i     | n-i-t-s-i | y-t-s-i   | y-t-s-i   | y-t-s-i
3p  | ŋ-i-m-i   | t-s-i-k-i | k-i-m-i   | s-a-m-i   | s-a-m-i | n-a-m-i   | n-a-t-s-i-m-i | n-i-m-i   | m-i       | m-i       | m-i
```

```
free forms:  u i y ŋ k d n m a s t
```


Combined Learning & Segmentation

Current Algorithm: segmentation based on form *and* meaning

```
while True:
    analysis=paradigm.analysis()
    candidates=SegmentationCandidates(analysis)
    best_candidate=candidates.select(key=rate_candidate,n=1)
    if not has_minimum_quality(best_candidate):
        break
    ... (do actual segmentation)
```

1. **Learn** the invariant meaning for all freely occurring forms
2. **Count**
   - how many of the cells that are implied by the learned meaning contain (a) the form, (b) a proper superstring, (c) neither of them
   - idem for all possible generalizations *n* of the meaning, restricted to added cells: a_n=0, b_n, c_n
3. **Choose** the best candidate by maximizing (in this order):
   $\frac{a}{a+b+c} - \frac{a+b}{a+b+c}$, $\frac{a_n+b_n}{a_n+b_n+c_n}$, $a_n + b_n + c_n$, and a+b+c
4. **Stop** if $\frac{a}{a+b+c} - \frac{a+b}{a+b+c} > 0.1$ and $\frac{a_n+b_n}{a_n+b_n+c_n} = 1$ doesn't apply to it

Combined Learning & Segmentation: Candidate evaluation

Current Algorithm: segmentation based on form *and* meaning

```
    | 1sg | 1pl | 2sg | 2pl | 3sg | 3pl | x
----+-----+-----+-----+-----+-----+-----+----
x   | !   | a   |     |     | a   | a   |
       [-1,-2,-3]A->P[-2,-x]   25%, (100%), 0, 4

    | 1sg | 1pl | 2sg | 2pl | 3sg | 3pl | x
----+-----+-----+-----+-----+-----+-----+----
x   | !   | a   |     |     | a   | a   | +!
       [-1,-2,-3]A->P[-2]      25%, 100%, 1, 4

    | 1sg | 1pl | 2sg | 2pl | 3sg | 3pl | x
----+-----+-----+-----+-----+-----+-----+----
x   | !   | a   | +!  | +!  | a   | a   |
       [-1,-2,-3]A->P[-x]     25%, 100%, 2, 4

    | 1sg | 1pl | 2sg | 2pl | 3sg | 3pl | x
----+-----+-----+-----+-----+-----+-----+----
x   | !   | a   | +!  | +!  | a   | a   | +!
       A[-1,-2,-3]            25%, 100%, 3, 4  WINNER
```

Learning & Segmentation

Segmentation based on form *and* meaning

- Necessary features as preliminary meaning of a marker
- Possible subsets (generalizations) as search space for segmentation candidates
- Changes in form and meaning are balanced:
  - ratio of the number of segmentable cells to the number of added cells
  - stop of the segmentation if the accuracy of the marker meaning isn't improved by the best candidate
  - additional possibilities: e.g. homonyms (specialization of markers, splitting up)
- Local optimization, bias, competition, mutual exclusive segmentations

Learning & Segmentation: Results

Segmentation based on form *and* meaning

1s	ŋu
1de	tsuku
1pe	ku
1di	tsi
1pi	di
2s	na
2d	tsi
2p	ni
3s	
3d	tsi
3p	mi

	1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s	-	-	-	-	-	ni	ni-tsi	nini	u	u-tsi	umi
1de	-	-	-	-	-	tsuku	tsuku	tsuku	tsuku	tsuku	tsuku
1pe	-	-	-	-	-	ku	ku	ku	ku	ku	ku
1di	-	-	-	-	-	-	-	-	tsi	tsi	tsi
1pi	-	-	-	-	-	-	-	-	i	i-tsi	i
2s	ŋi	tsi-ki	ki	-	-	-	-	-	na	na	na
2d	ŋi-tsi	tsi-ki	tsi-ki	-	-	-	-	-	tsi	tsi	tsi
2p	ŋini	ki-ni	ki-ni	-	-	-	-	-	ni	ni	ni
3s	ŋi	tsi-ki	ki	tsi-ki	sa	na	na	nimi	y	y-tsi	y-mi
3d	ŋi-tsi	tsi-ki	ki-ni	sa	sa-mi	na-tsi	na-tsi	ni-tsi	y-tsi	y-tsi	y-tsi
3p	ŋi-mi	tsi-ki	ki-mi	sa-mi	sa-mi	na-mi	na-tsi-mi	ni-mi	mi	mi	mi

Learning & Segmentation: Candidate Evaluation

Segmentation based on form *and* meaning

		1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s		1s							+!			+!	
1de		1de											
1pe		1pe							+			+	
1di	tsi	1di									tsi	tsi	tsi
1pi		1pi										+!	
2s		2s		+!								+	
2d	tsi	2d	!	!	!						tsi	tsi	tsi
2p		2p		+								+	
3s		3s		+!		+!			+			+!	
3d	tsi	3d	!	!				!	!	!	!	!	!
3p		3p		+!		+			+!			+	

SA[-sg,+du,-pl] ==> SAP[-sg,+du,-pl] 36.67%, 52.94%, 17, 30

		1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s		1s										
1de		1de										
1pe		1pe										
1di	tsi	1di									tsi	tsi	tsi
1pi		1pi											
2s		2s											
2d	tsi	2d	!	!	!						tsi	tsi	tsi
2p		2p											
3s		3s											
3d	tsi	3d	!	!				!	!	!	!	!	!
3p		3p											

SA[-sg,+du,-pl] 36.67%, (100%), 0, 30 WINNER

Learning & Segmentation: Candidate Evaluation

Segmentation based on form *and* meaning

		1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s		1s										
1de		1de										
1pe		1pe										
1di		1di										
1pi		1pi										
2s		2s					!	ki				
2d		2d					ki	ki				
2p		2p					+!	+!				
3s		3s					!	ki				
3d		3d					ki	!				
3p		3p					+!	+!				

[-1,-pl]A->P[+1,-2,-3,-sg] ==> P[+1,-2,-sg] 37.5%, 100%, 4, 8 WINNER

		1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s		1s										
1de		1de										
1pe		1pe										
1di		1di										
1pi		1pi										
2s		2s										
2d		2d										
2p		2p										
3s		3s								y	!	!
3d		3d								y	y	y
3p		3p										

[-1,-2,+3,-pl]A->P[-1,-2,+3] 33.33%, (100), 0, 6 WINNER

Learning & Segmentation: Candidate Evaluation

Segmentation based on form *and* meaning

		1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s		1s										
1de		1de										
1pe		1pe										
1di	+!	1di								+!	+!	+!
1pi	+!	1pi								i	i	i
2s		2s										
2d	+!	2d	+!	+!	+!					+!	+!	+!
2p	+!	2p	+!	+!	+!					+!	+!	+!
3s		3s										
3d		3d										
3p		3p										

[+1,+2,-3,-sg,-du,+pl]A->P[-1,-2,+3] ==> SA[+2,-3,-sg] 0%, 100%, 19, 3

		1s	1de	1pe	1di	1pi	2s	2d	2p	3s	3d	3p
1s		1s					ni	ni	!			
1de		1de										
1pe		1pe										
1di		1di										
1pi		1pi										
2s		2s										
2d		2d										
2p	ni	2p	!	ni	ni					ni	ni	ni
3s		3s							!			
3d		3d			ni				ni			
3p		3p							ni			

SAP[-3,-du] & SAP[-1] 6.25%, 100%, 0, 48 STOP

Learning & Segmentation: Outlook

Segmentation based on form *and* meaning

- Homonymy as 'last resort' for learning
- Homonymy for the residiue of a cell where a marker was segmented (syncretism as an option)
- Account for the effects of a segmentation on other markers
- Account for possible blocking of markers in a single cell
- Consider marker type, marker length, frequency, etc.?
- Invert the algorithm (find markers for given meaning)?

Conclusion

Automatic paradigm analysis...

- helps to find patterns and represent them
- allows to quantify & compare
- demands to develop numerical criteria for the quality of analyses
- makes it possible to compare analyses

Bank, Sebastian / Henze, Daniela (2010). Intersecting Multiargument Feature Specifications. In: Sebastian Bank, Doreen Georgi & Jochen Trommer (eds.), *2 in Agreement. Linguistische Arbeits Berichte 88*, Universität Leipzig, 247-265.

Lahaussois, Aimée (2002). *Aspects of the grammar of Thulung Rai: an endangered Himalayan language*. Ph.D Dissertation University of California, Berkeley. Available online: http://halshs.archives-ouvertes.fr/halshs-00004761

Pertsova, Katya (2007). *Learning Form-Meaning Mappings in Presence of Homonymy*. PhD thesis: University of California, Los Angeles.

Tamura, Suzuko (2000). *The Ainu Language*. Tokyo: Sanseido.