The “Jump and Stay” Method
to Discover Proper Verb Centered Constructions in Corpus Lattices

Sass Bálint

MAGYAR TUDOMÁNYOS AKADÉMIA
NYELVTUDOMÁNYI INTÉZET

Research Institute for Linguistics, Hungarian Academy of Sciences
`sass.balint@nytud.mta.hu`

previous work

verb centered construction = VCC
proper VCC = pVCC
theoretical model:
double cubes + corpus lattices

current contribution

effective implementation
a verbal constr. discovery method
the “jump and stay” principle
preliminary eval. on Hungarian data

conclusion

1 **corpus lattice – interesting**
it refers to the location of pVCCs, worth to investigate more closely, using our implementation

2 **“jump and stay” – promising**
discovers pVCCs in corpus lattices, can be considered a baseline

1 pVCC

VCC = verb + slots + fillers
slot = PP/NP deps (incl. subject)
***proper* verb centered construction (pVCC):**

complete = contains all necessary elements
clean = does not contain any unnecessary element
- free slots = complements
- fillers = idiomatic

a MWE: take part
a pVCC: take + SBJ + OBJ:part + in

free slots + filled slots (complementation + collocation) are equally important

This concept of completeness is essential (and unique) here.

– *Why are pVCCs important?*

pVCCs = different meanings / usage patterns of verbs.
Idea: a **dictionary** should present *exactly* the set of pVCCs concerning a verb. We handle all of them uniformly, in one framework.

2 Initial Model

basic unit: *clause*

representation of a clause:
double cube (DC)

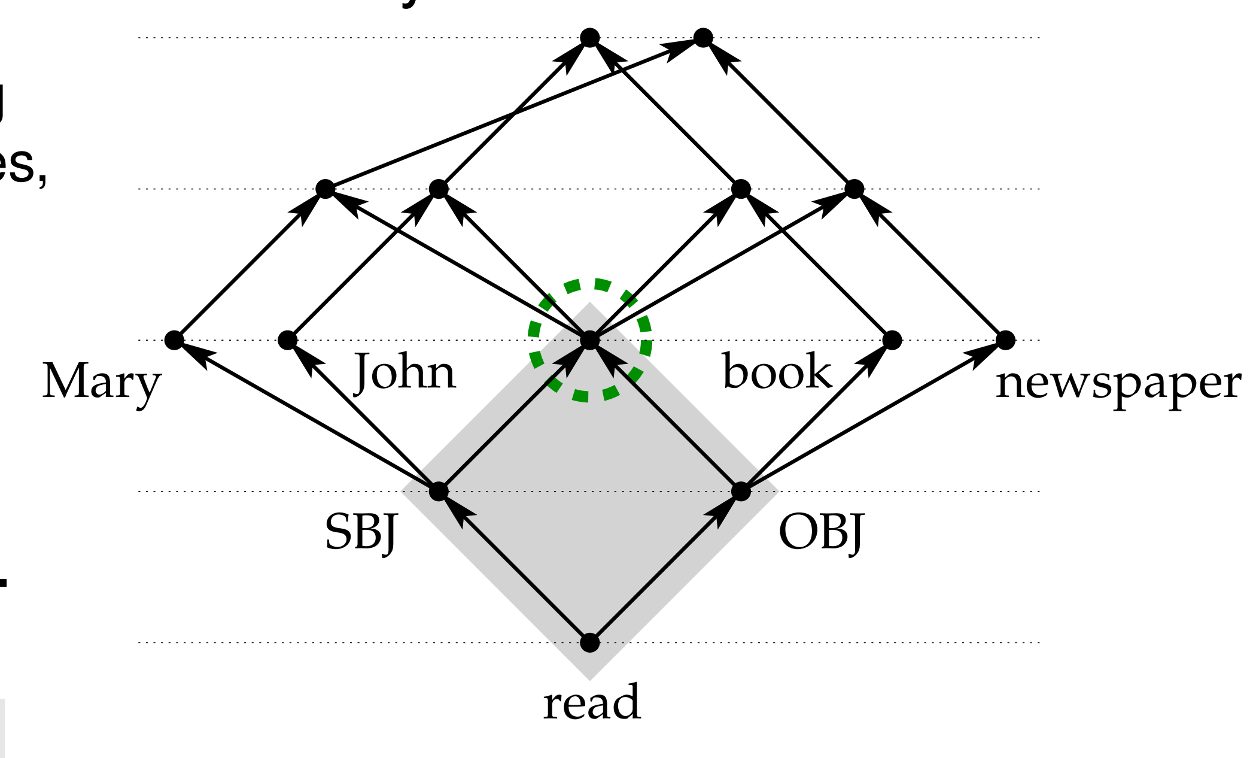
representation of a corpus:
corpus lattice (CL)
created from DCs containing the *same* main verb using a *lattice combination* operation ($\oplus$)

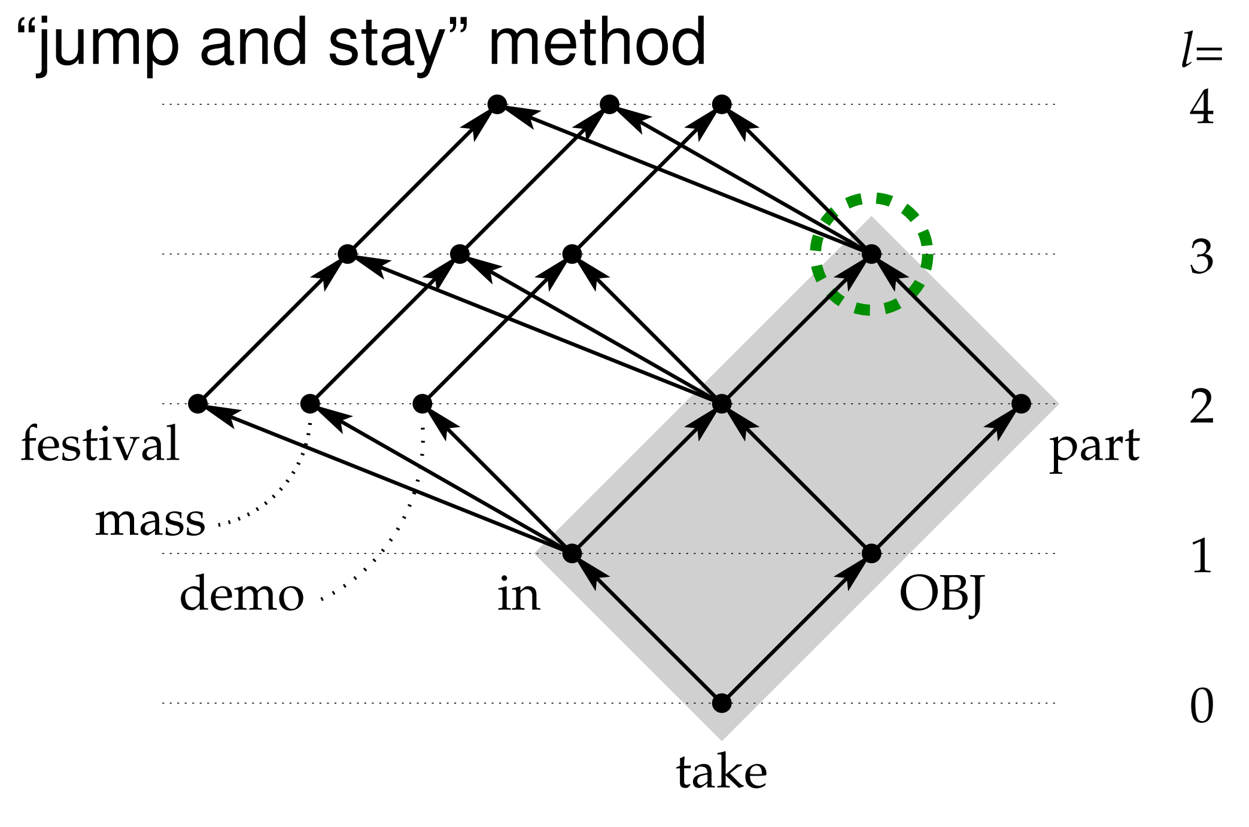
$\rightarrow$ a CL represents all clauses of a given verb, and also **the distribution of all free and filled slots occurring beside this verb.**

What?

input **corpus**		output **pVCCs**
John reads a book.	He takes part in a demo.	read + SBJ + OBJ
Mary reads a newspaper.	He takes part in a mass.	take + SBJ + OBJ + into:account
He took this opinion into account.	He takes part in a festival.	take + SBJ + OBJ:part + in
He will take that info into account.	...	...

How? analysed clauses $\rightarrow$ DC $\rightarrow$ CL $\rightarrow$ “jump and stay” method





read SBJ:John OBJ:book $\oplus$
read SBJ:Mary OBJ:newspaper

three three-dimensional DCs $\oplus$-ed in two dimensions
(pVCCs are circled. $f > 1$ in the gray areas.)

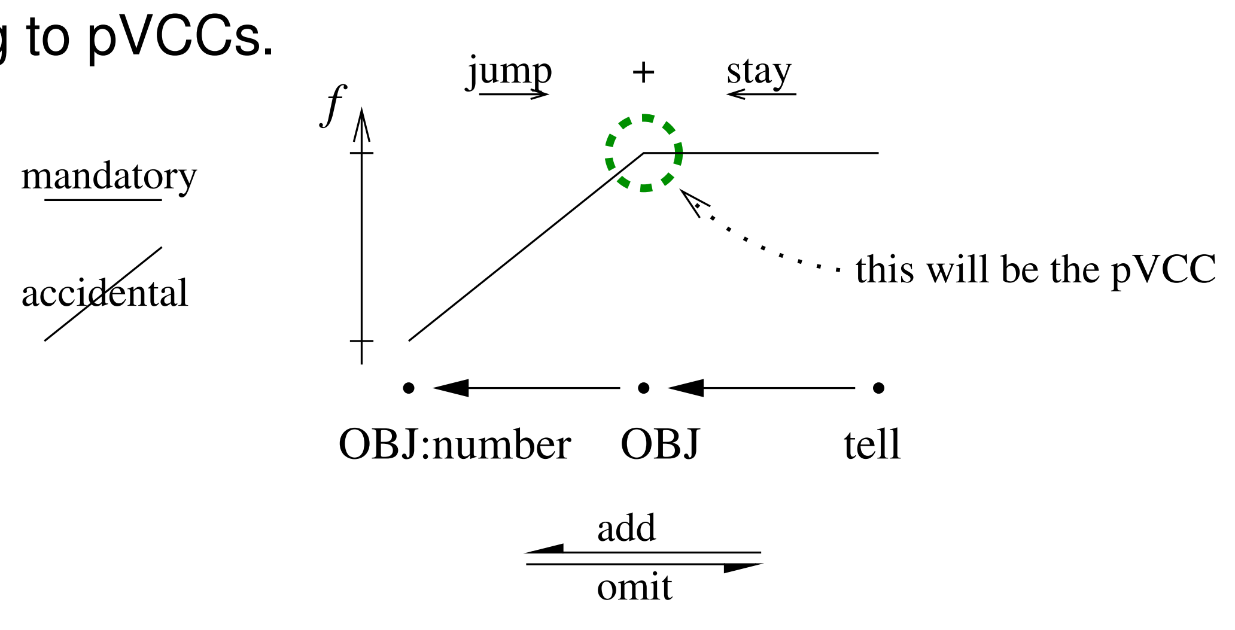
Where? `https://github.com/sassbalint/double-cube-jump-and-stay`

3 “Jump and Stay”

$f(v)$ = corpus frequency of the VCC represented by vertex v.

Observation: pVCC vertices can be characterized as...
- going top-down f substantially increases
- better if located higher

The value of f jumps up and then stays the same at certain locations pointing to pVCCs.



The “jump and stay” principle:

- **jump** = step to an adjacent vertex downwards in the CL if f substantially increases
- **stay** = step to an adjacent vertex upwards in the CL if f remains roughly the same

the “jump and stay” idea is *nicely consistent* with the fact that constructions have mandatory and accidental elements.

jump = **omit accidental element**
stay = **add mandatory element**

a typical pVCC is *an endpoint of both jumps and stays*

stays increase completeness
jumps increase cleanness

4 Model Impl.

important: to be able to effectively step from a vertex to an adjacent one
solution: store vertices and edges in hashes

5 Data

format: a specific JSON. It can be generated from a shallow parsed input corpus:
verb + slots + fillers need to be identified.

data: 28 million analysed Hungarian clauses, 7% dev + 93% test

How does our algorithm work in practice?

```
#4                                                  f=  l=
["FAC", null]                                       309  1
 A stay found, we follow.
["FAC", null, "NOM", null]                          309  2
 A stay found, we follow.
["FAC", "jó", "NOM", null]                          307  3
 A stay found, we follow.
["ACC", null, "FAC", "jó", "NOM", null]             300  4
 No stay (ratio=5.17 > 1.7), we stop.
 No appropriate jump (keeping a filler, 1.02 < 4), we stop.
["ACC", null, "FAC", "jó", "NOM", null]             300  4  pVCC

#22699                                              f=  l=
["ACC", "költségvetés", "FAC", "jó", "NOM", null] 4  5
 No stay (ratio=2.00 > 1.7), we stop.
 An appropriate jump (keeping a filler, 4<) found, we follow.
["ACC", null, "FAC", "jó", "NOM", null]             300  4
 No stay (ratio=5.17 > 1.7), we stop.
 No appropriate jump (keeping a filler, 1.02 < 4), we stop.
["ACC", null, "FAC", "jó", "NOM", null]             300  4  pVCC
```

hagy + NOM + ACC + FAC:jó $\leftarrow$ pVCC
allow + SBJ + OBJ + FAC:good (= approve + SBJ + OBJ)

6 Algorithm

```
1. take each vertices of the CL
2. omit some: too long ($l > 8$), too rare ($f < 3$), no out-edge
3. look for a stay:
     if $f(\text{actual})/f(\text{above}) < 1.7$
     → this a stay
     → step to the vertex above
4. no stay? look for a jump:
     if $f(\text{below})/f(\text{actual}) > 4$
     → this a jump
     → step to the vertex below
5. a new vertex reached?
     → repeat steps 3. and 4.
6. if no stay and no jump can be found → stop
   if the current VCC is not at the top of the CL → it is a pVCC
```

in step 4: no jump if it would omit the *last* filler from a VCC

7 Evaluation

The algorithm was run on two verbs: húz (draw/pull) and vet (cast/throw). Then the first 20 pVCCs (according to f value) was investigated whether they are correct or not.

Results

70-80% of the pVCCs are perfect.

one single real error (2.5%): **#17** – filler vonal (line) is missing.

Discussion

- many complete + clean pVCCs
- different pVCCs are often translated using different verbs
- optionality: #2 and #20
- our concept of completeness: #28/#29/#30.
  a certain filler $\rightarrow$ a new complement $\rightarrow$ a new pVCC
- interference: #24 and #26

8 Future Work

- handling pronouns
- better threshold values
- *what to do when a few elements seem to be mutually exclusively mandatory at a point?*
  take into:account/consideration
- application for other languages and other structures

Acknowledgement. This research was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences (case number: BO/00064/17/1; duration: 2017-2020).

#	?	Hungarian pVCC	f	word by word	English counterpart
		húz	**9505**	**draw/pull**	
1.	✓	ACC	8304	OBJ	pull sg
2.	✓	**ACC:idő**	**420**	**OBJ:time**	**temporize**
3.	✓	ACC:haszon + ELA	412	OBJ:profit + from	profit from sg
4.	✓	ACC + SUB:maga	239	OBJ + onto:oneself	put sg on
5.	✓	ACC + után:maga	209	OBJ + after:oneself	pull sg behind oneself
6.	✓	ACC + ALL:maga	207	OBJ + to:oneself	pull/draw sy to oneself
7.	≈	ACC + SUB:fej	199	OBJ + onto:head	put sg on one’s head
8.	✓	felé	169	towards	be drawn/attracted towards sg
9.	✓	ACC:rövid	166	OBJ:short	get the worst of it
10.	✓	ACC:vonal	152	OBJ:line	draw a line
11.	✓	ACC:láb	139	OBJ:foot	drag one’s feet
12.	✓	ACC:ujj + INS	118	OBJ:finger + with	pick a quarrel with sy
13.	p	ACC + NOM:aki	108	OBJ + SBJ:who	who pulls sg
14.	p	ACC + TEM:az	107	OBJ + at:that	pull sg at that time
15.	✓	ACC + INS:maga	92	OBJ + with:oneself	drag sy/sg with oneself
16.	✓	ACC + felé	85	OBJ + towards	pull sg towards sg
17.	×	**ACC + közé**	**82**	**OBJ + between**	**draw sg *(a line)* between sg**
18.	✓	ACC:szék	80	OBJ:chair	draw one’s chair up
19.	✓	ACC:határ	77	OBJ:border	set limits
20.	✓	**ACC:idő + INS**	**77**	**OBJ:time + with**	**temporize on sg**
		vet	**14759**	**cast/throw**	
21.	✓	ACC	13649	OBJ	cast/throw sg
22.	≈	ACC + SUB	5437	OBJ + onto	cast/throw sg on sg
23.	✓	ACC:vég + DAT	2632	OBJ:end + for	put an end to sg
24.	✓	**ACC + SUB:szem**	**1085**	**OBJ + onto:eye**	**reproach sy for sg**
25.	≈	ACC:maga	964	OBJ:oneself	throw oneself
26.	✓	**ACC:pillantás + SUB**	**839**	**OBJ:glance + onto**	**glance at sy/sg**
27.	✓	ACC + SUB:papír	673	OBJ + onto:paper	note down sg
28.	✓	**ACC:fény + SUB**	**402**	**OBJ:light + onto**	**reflect *(well/badly)* on sy/sg**
29.	✓	**ACC:szám + INS**	**371**	**OBJ:number + with**	**take sg into account**
30.	✓	**ACC:gát + DAT**	**362**	**OBJ:obstacle + for**	**put a stop to sg**
31.	≈	ACC:maga + SUB	345	OBJ:oneself + onto	throw oneself into sg
32.	✓	ACC:maga + ILL	339	OBJ:oneself + into	throw oneself into sg
33.	p	ACC:az + SUB:szem	302	OBJ:that + onto:eye	reproach sy for that
34.	✓	SUB:maga	297	onto:oneself	have only oneself to blame
35.	✓	ACC:szem + SUB	285	OBJ:eye + onto	take a fancy to sy/sg
36.	✓	ACC:kereszt	261	OBJ:cross	cross oneself
37.	✓	ACC:árnyék + SUB	258	OBJ:shadow + onto	cast/throw a shadow over sy/sg
38.	✓	ACC + ILL:lat	240	OBJ + into:*lat*	use sg *(one’s power)*
39.	p	ACC + SUB:én	225	OBJ + onto:me	cast/throw sg onto me
40.	p	ACC + NOM:aki	201	OBJ + SBJ:who	who casts/throws sg