Identifying constructicon entries in query text

ICCG13
26–28 August 2024

Bálint Sass

Hungarian Constructicon project – OTKA K 147452

HUN-REN Hungarian Research Centre for Linguistics
Institute for Lexicology
sass.balint@nytud.hu

outline

1. a view on constructions
2. motivation
3. representation
4. the algorithm for identification
5. examples
6. demo
7. availability

a simple and radical view on constructions

- construction (cxn) $\stackrel{\text{def}}{=}$ learned pairing of form and meaning ✓

  words are cxns! (Goldberg, 2006; Hilpert, 2014)

  cxns are combinations of slots and fillers
  $\rightarrow$ a continuum:
  words … expressions with open slots … constructional schemas

- constructicon (ccn) $\stackrel{\text{def}}{=}$ inventory of cxns

  a dictionary-like lexical resource
  which has **head-cxns** instead of headwords

  a *superset* of the dictionary as it contains all words as well

  "covers everything" since "it's constructions all the way down"

motivation

1. *project:* we are building a ccn for Hungarian
2. *aim:* **make it accessible to anyone**
   $\rightarrow$ search by giving arbitrary short text as query
3. *approach:* **analyse** query text, **extract** cxns
   and display their ccn entries to the user
4. *question:* how to identify cxns in the query text?

remark: the title of my talk is not accurate terminologically

ccn entry $\stackrel{\text{def}}{=}$ the cxn itself as head-cxn + its meaning + other info

$\rightarrow$

Identifying constructicon entries in query text ✗

Identifying head-constructions in query text ✓

a generalization of the online dictionary UI

1. **online dictionary UI:** (1) one-word query text, (2) consider it as a dictionary headword, (3) search for it in the dictionary database
   (the "identification" of the headword in the query text is trivial in this case)
2. *generalization to multiple headwords:*
   a dictionary UI *could* handle multiple headwords, taking the query word by word as potential headwords – e.g. *in school*
3. *generalization to constructions*, working with a ccn:
   head-cxns are often multiword
   → allow multiword query text
   a multiword query often contains more than one cxn **intertwined**
   → search for all of them
4. **ccn UI:** (1) arbitrary query text, (2) identify the head-cxns in it, (3) search for all of them in the ccn

representation: filler-slot trees

- we mentioned: cxns are combinations of slots and fillers
- we can consider that *each filler is in a slot*
  take a look at SLOT $\leftrightarrow$ *have SLOT at his disposal* $\leftrightarrow$ *book*
- *consequence:* a tree-representation seems suitable
  for representing cxns: **slot = edge; filler = node** $\rightarrow$ filler-slot trees
- dependency trees are similar $\leftarrow$ we will use them

  what kind of transformations would we need
  to convert dependency trees into filler-slot trees?
  $\leftarrow$ we leave this question open for now

  examples:

  English prepositions will be slots

  Hungarian case markers will be slots

a question about the representation

Is the filler-slot tree representation capable of representing any cxns?

a question about the representation

Is the filler-slot tree representation capable of representing any cxns?

I assume, it is.

If you can support this position, or
if you can think of a cxn that is problematic to represent in this way,
please share it with me!

using the representation

- filler-slot trees for representing cxns – *with open slots*
  $\rightarrow$ cxns are stored in the ccn in this form
- filler-slot trees for representing query text – *no open slots*
  $\rightarrow$ query text is analysed on the fly as entered

- identifying cxns in the query text
  by matching representations of the head-cxns
  onto the representation of the query text

the algorithm

1. start from the root node of the tree of the query text;
2. taking the ccn, search for head-cxns which **matches** here;
3. take the maximal/longest matching head-cxn if there are several;
4. **remove** the found head-cxn from the tree;
5. after the removal the tree falls apart into smaller trees;
6. take these smaller trees one by one and start over the algorithm.

$\rightarrow$ the identified cxns will correspond to
different fragments of the filler-slot tree of the query text

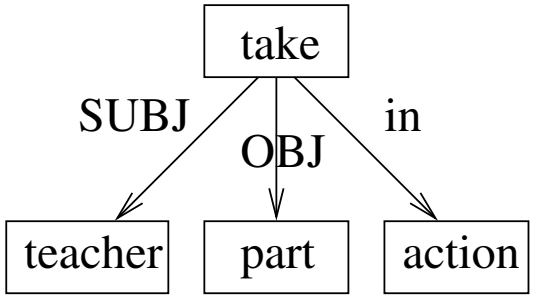
how the algorithm works

query text: *the teacher takes part in the action*

how the algorithm works

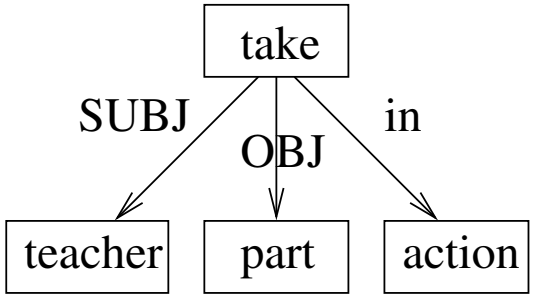
example #1

query text: *the teacher takes part in the action*

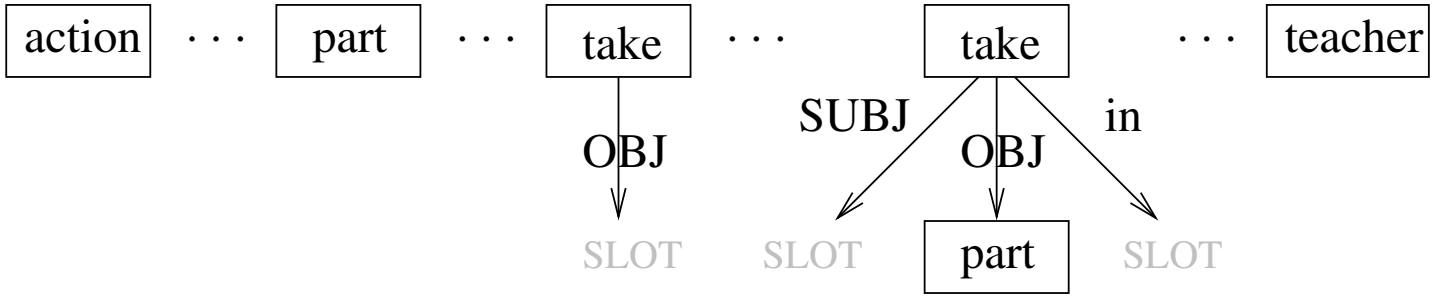


how the algorithm works

query text



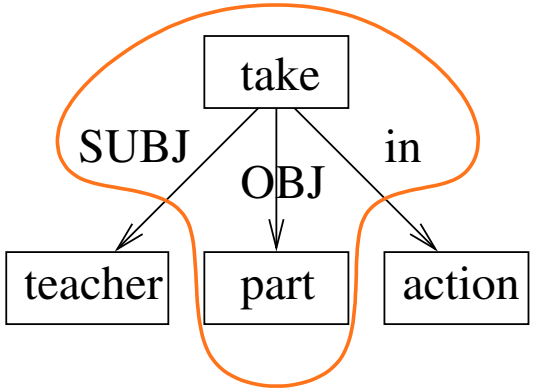
head-cxns



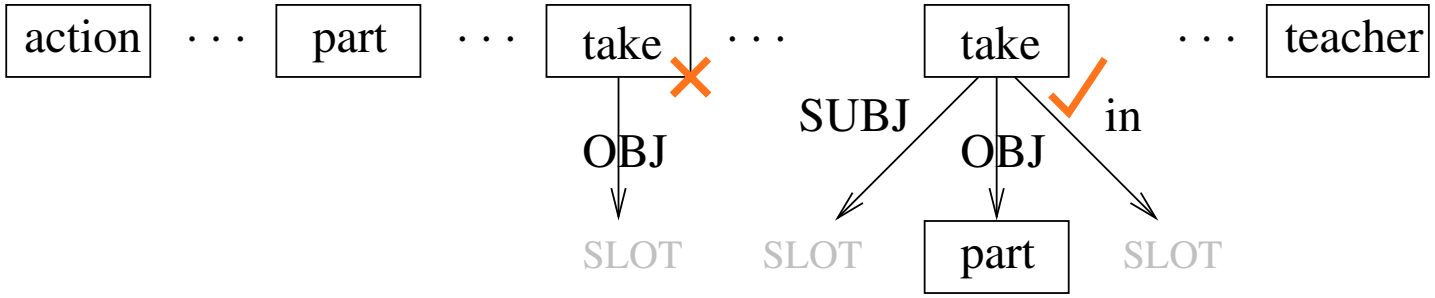
how the algorithm works

example #1

query text



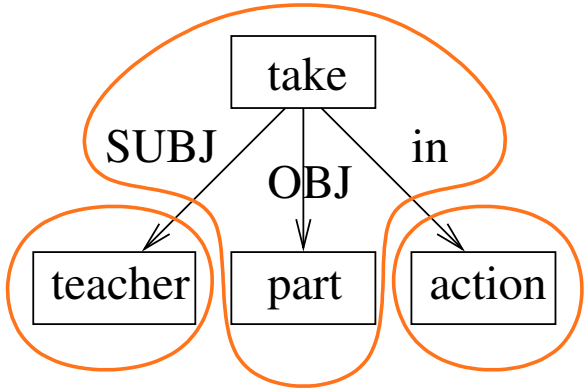
head-cxns



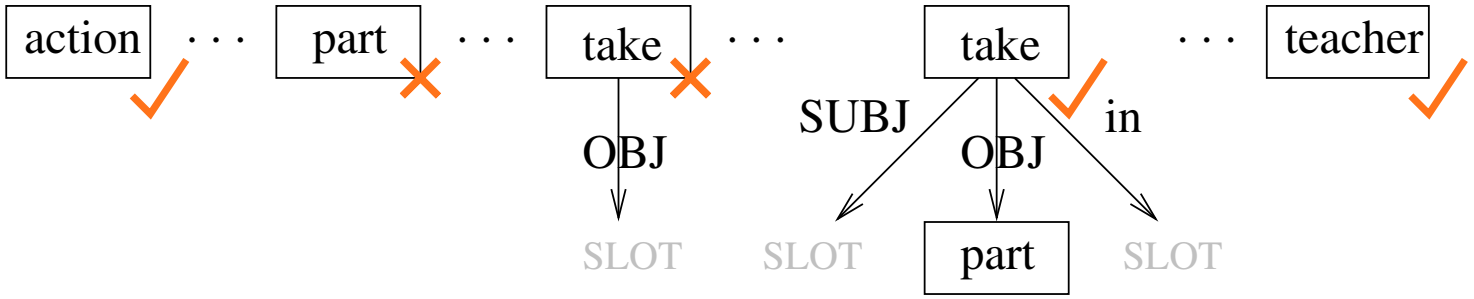
how the algorithm works

example #1

query text



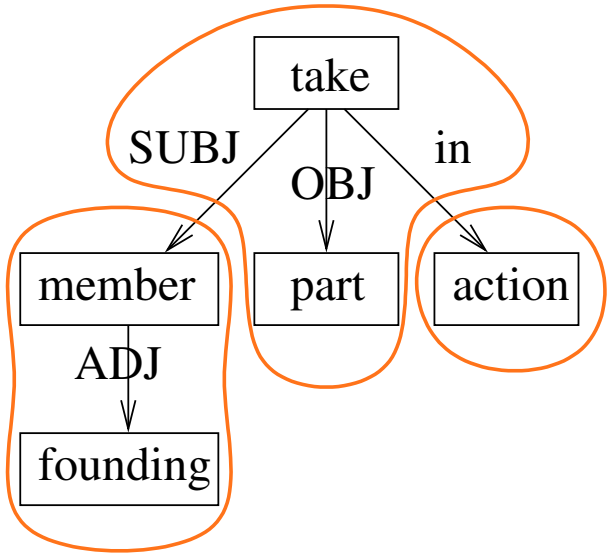
head-cxns



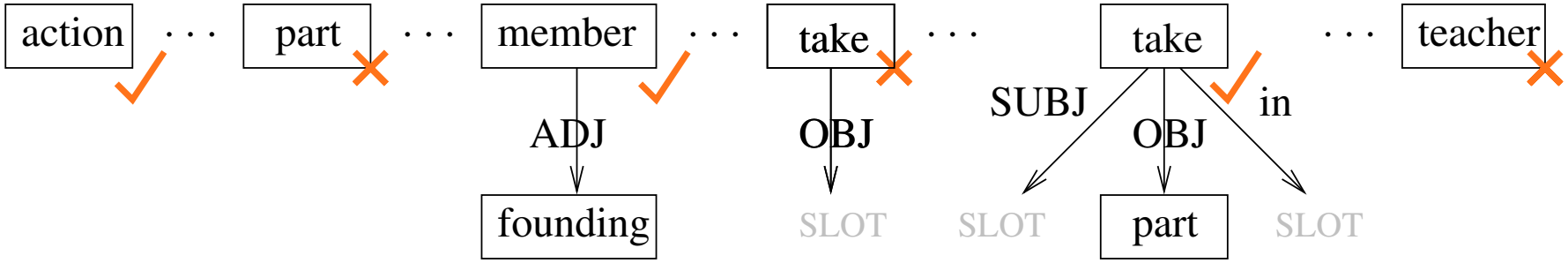
how the algorithm works

example #2

query text: *the founding member takes part in the action*



head-cxns



demo

`https://ccn-dev.nytud.hu`

Examples in Hungarian.

availability

https://ccn-dev.nytud.hu

username: `dev`
password: `fortesting`

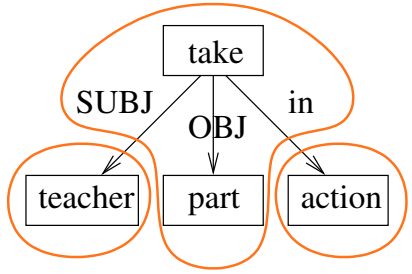
Will be integrated into **`https://ccn.nytud.hu`**
to be opened this year.

Feel free to try a few queries and
contact me if you have any questions or comments.

Bálint Sass
`sass.balint@nytud.hu`

summary

1. a view on constructions — words are cxns → ccn "covers everything"
2. motivation — search a ccn using arbitrary text
3. representation — filler-slot trees
4. the algorithm for identification — match and remove
5. examples



6. demo
7. availability — **https://ccn-dev.nytud.hu** dev/fortesting

Bálint Sass
sass.balint@nytud.hu

acknowledgement

The research that is the subject of this presentation was supported by the OTKA (K 147452) grant of the National Research, Development and Innovation Office (NKFIH), Hungary. Project K 147452 is being implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the K_23 funding scheme.

Identifying constructicon entries in query text

ICCG13
26–28 August 2024

Bálint Sass

Hungarian Constructicon project – OTKA K 147452

HUN-REN Hungarian Research Centre for Linguistics
Institute for Lexicology
sass.balint@nytud.hu