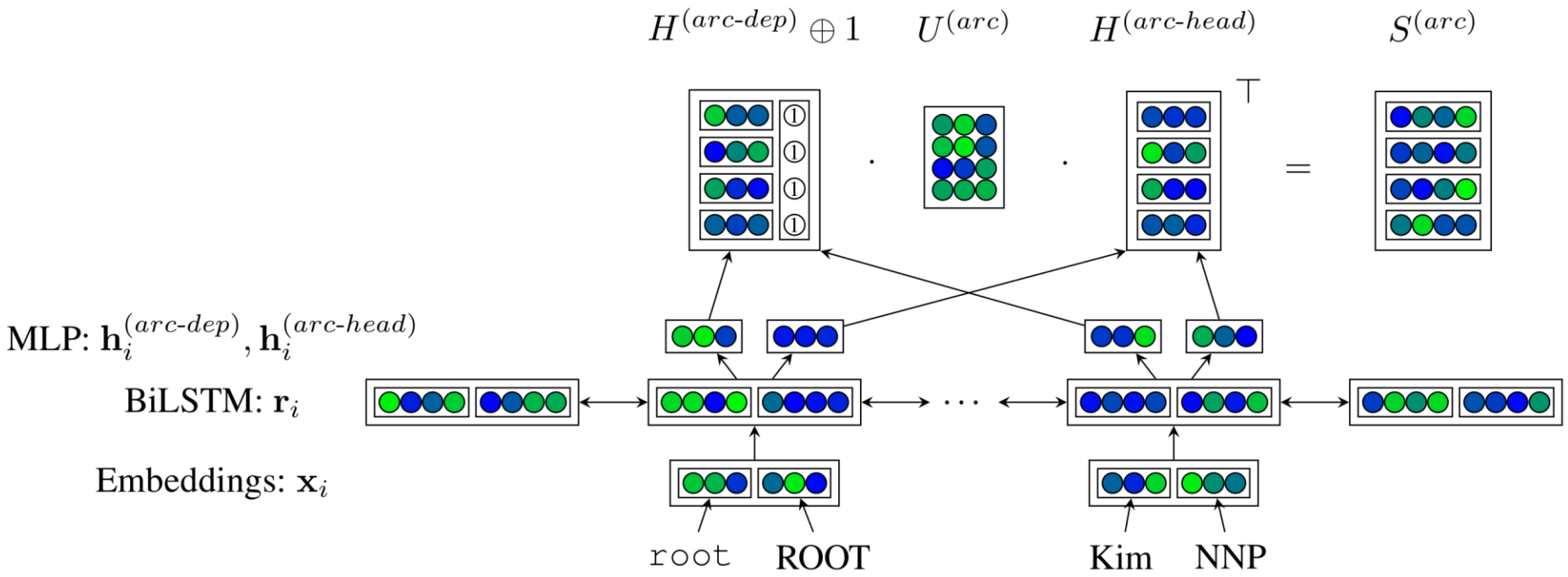
Neural Span-based *MRS Parsing

Jan Buys

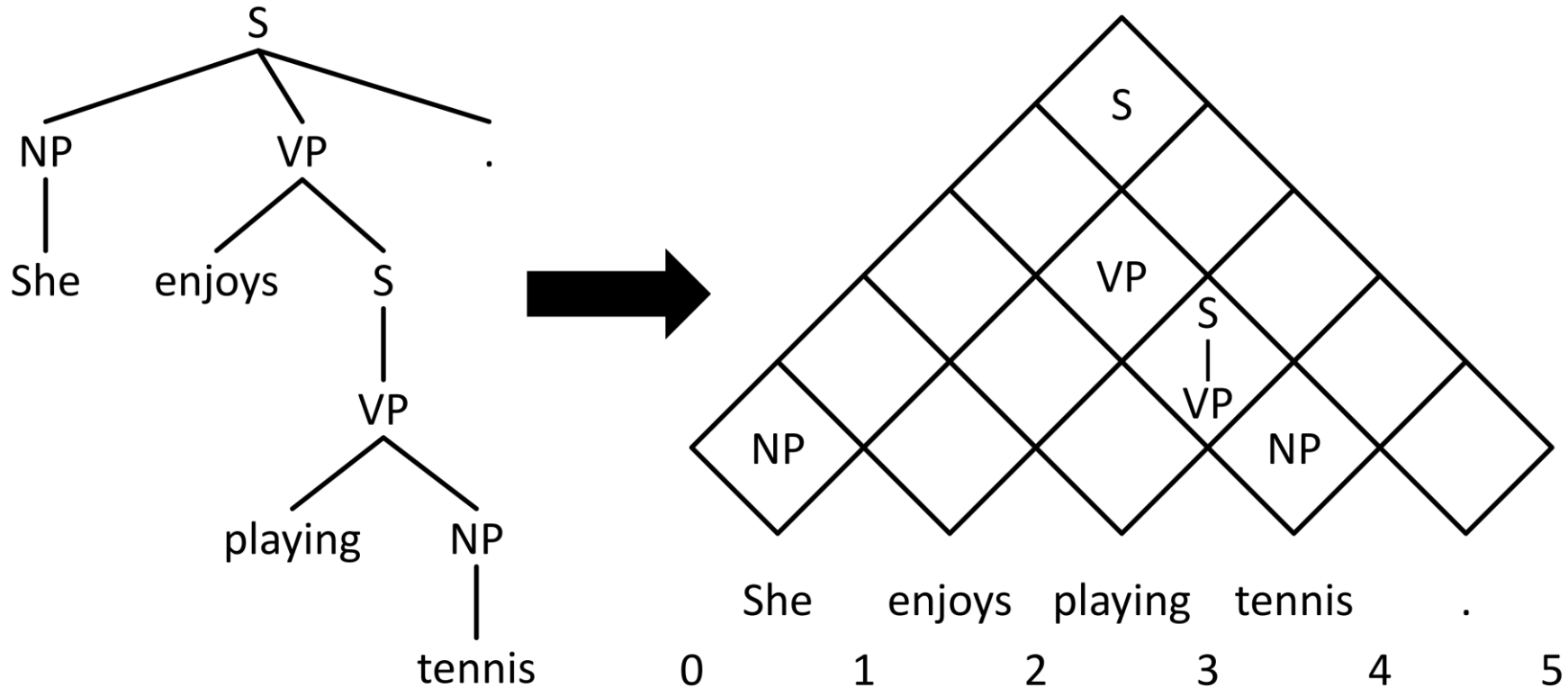
University of Cape Town

Neural Span-based Parsing



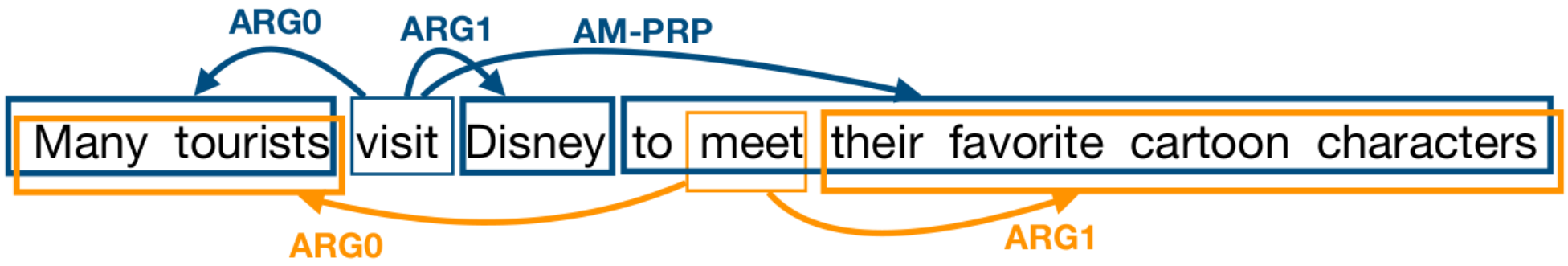
- Minimal feature-based Dependency Parsing (Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017)

Neural Span-based Parsing



- Span-based Constituency Parsing (Cross and Huang, 2016; Stern et al., 2017)

Neural Span-based Parsing



- Span-graphs for Semantic Role Labelling (He et al., 2018; Peng et al., 2018)

Neural Span-based Parsing

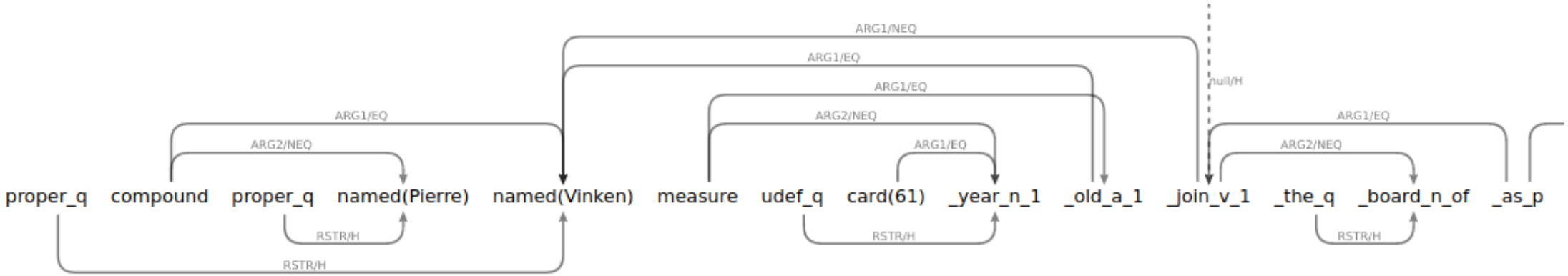
Task	Spans annotated with labels
NER	Barack Obama (person) was born in Hawaii (location).
Consti.	And their suspicions (NP) of each other (NP) run deep (ADVP). PP: of each other; VP: run deep; NP: their suspicions of each other; S: And their suspicions of each other run deep.

Task	Spans and relations annotated with labels
RE	The burst (cause-effect) has been caused by pressure.
Coref.	I voted for Tom (coref.) because he is clever.
SRL	We (ARG0) brought (ARG2) you (ARG1) the tale of two cities.
OpenIE	The four lawyers (ARG0) climbed out (ARG1) from under a table.
Dep.	The (det) entire (amod) division (nsubj) employs (dobj) about (advmod) 850 (nummod) workers.

- Span-relation representations for multiple tasks (Jiang et al., 2020)

Neural Span-based *MRS Parsing

- Goal: Apply to parse sentences to EDS/DMRS graphs
- Challenge: computationally tractability



Pierre Vinken, 61 years old, will join the board as ...

Neural Span-based *MRS Parsing

- How exactly do we set up the problem and pre-process the data?
- Can we exploit constraints on the graph structure to make parsing more tractable?
- Can we make it easier to incorporate information or constraints from the grammar into a data-driven parser?

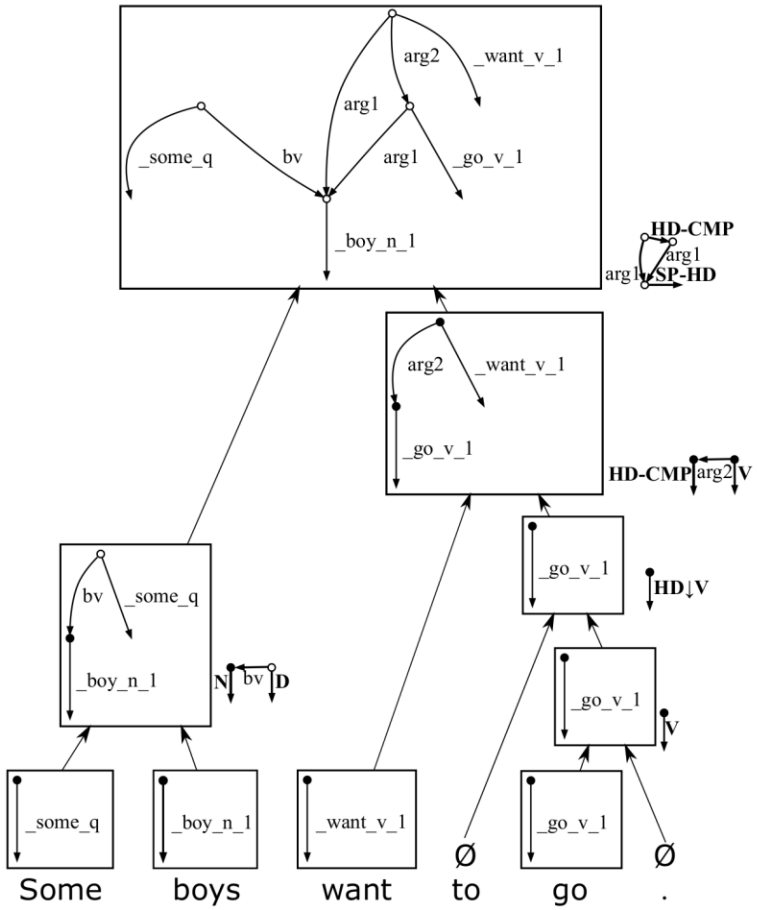
Related work: Neural *MRS Parsing

- Buys and Blunsom (2017): Transition-based *MRS parser

Action	Stack	Buffer	Arc added
init(1, person)	[ ]	(1, 1, person)	-
sh(1, every_q)	[(1, 1, person)]	(2, 1, every_q)	-
la(BV)	[(1, 1, person)]	(2, 1, every_q)	(2, BV, 1)
sh(2, _v_1)	[(1, 1, person), (2, 1, every_q)]	(2, 1, _v_1)	-
re	[(1, 1, person)]	(3, 2, _v_1)	-
la(ARG1)	[(1, 1, person)]	(3, 2, _v_1)	(3, ARG1, 1)

Related work: Neural *MRS Parsing

- Chen et al. (2018): Synchronous Hyperedge Replacement Grammar-based Semantic Parsing

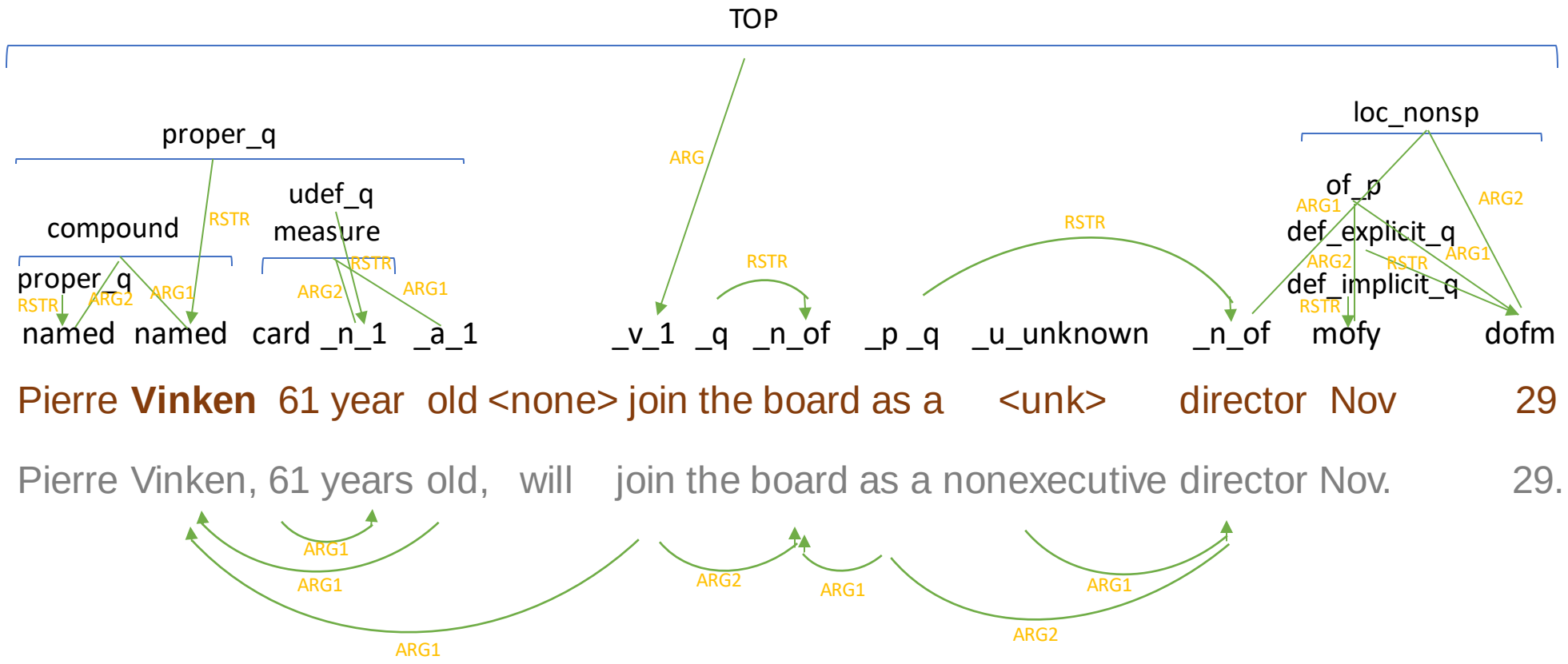


Neural Span-based *MRS Parsing

Caveats

- Work in progress
- Want to fix some issues in Buys and Blunsom (2017)
- This talk: mostly about representation and pre-processing
- Based on ERG, so may be English-specific

Span-based *MRS



Defining the Parsing Problem

Pre-processing:

- Tokenize input
- Map tokens to lemmas / CARGs / unknowns / (none)

Predict:

- Semantic labels of terminals (surface predicates, CARG nodes)
- Non-terminal semantic nodes as span labels – inside the syntax tree
  - Predict jointly with their edges as span-span relations (?)
- (other) Edges as relations between spans

Data

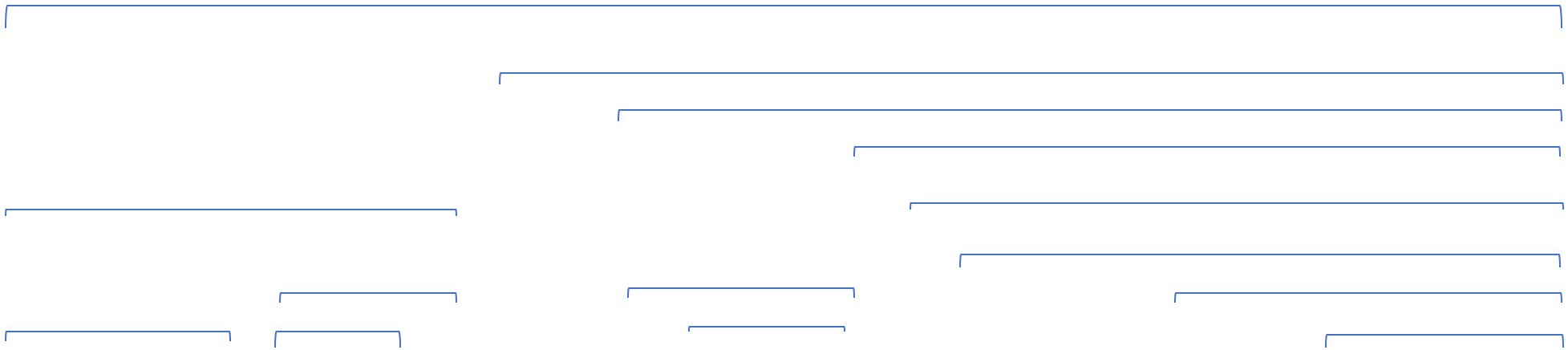
- Redwoods 1214
- Input using PyDelphin
- No other preprocessing tools (but some heuristics required)

Data Processing: Input Annotation Layers

- Tokenization lattice
- Syntactic derivation tree
- MRS -> convert to DMRS (small number of conversion errors)

Data Processing: Derivation tree

- Collapse unary chains
- Pre-terminal nodes map to one or more tokens (some preprocessing required)

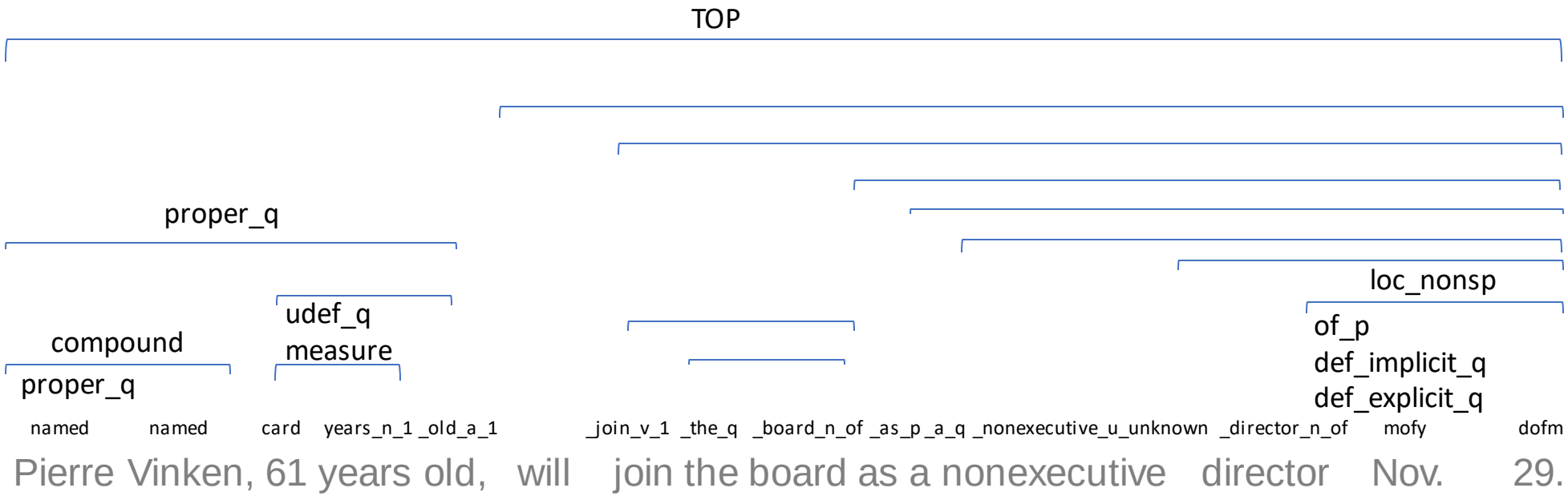


Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

Data Processing: Predicates

Map MRS predicates to derivation tree nodes

- Map character spans to token spans, match to tree node spans



Data Processing: Predicates

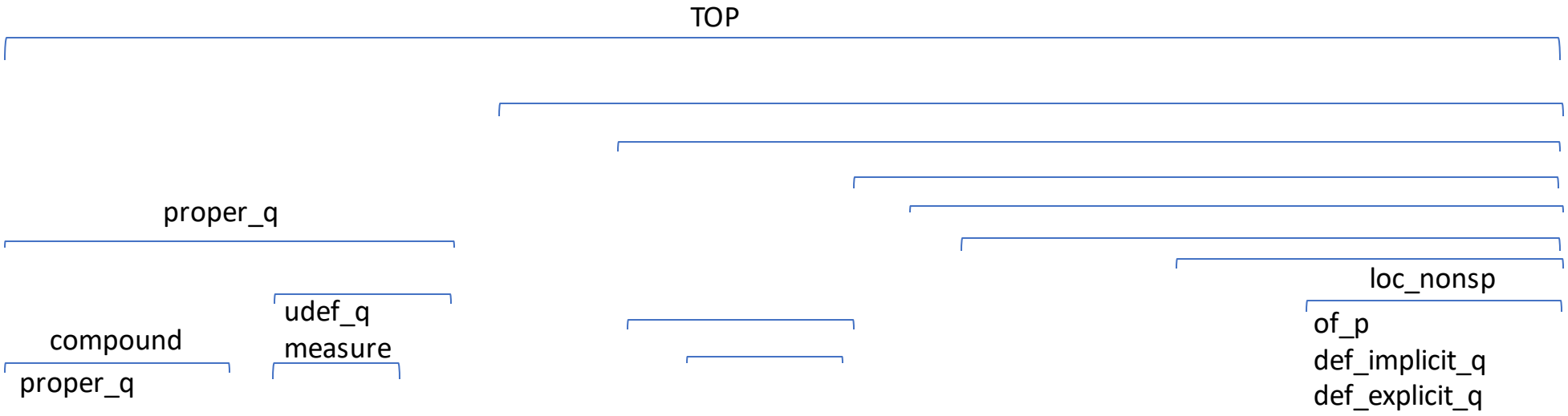
Map MRS predicates to derivation tree nodes

- Mismatch for small number of multi-word tokens
  - New York-based -> [New York-] [based]
  - $37-a-share -> [$ 37-] [a- share]
  - summer/winter rate differential collections -> [summer] [/ winter rate differential collections]

Data Processing: Surface predicates

Extract lemmas from predicates

- Same for nodes with CARGs (except for span with surface predicate + CARG)



Data Processing: Surface predicates

Some surface predicates align to multiple tokens – not preterminals

- Map them down to preterminal nodes, along with their direct ancestors with same span

```
(['udef_q', 'card', '_day_n_of']
   ([] 30- )
   ([] day ) )


(['udef_q', 'compound', 'udef_q', '_energy_n_1', '_service_n_1']
   ([] energy- )
   ([] services ) )
```

Data Processing: Surface predicates

Extract lemmas from predicates: special cases

- Tokens with multiple surface forms for a token: e.g. predicates for prefixes

  refile: _file_v_1 and _re-_a_again

  Uncomplaining: complain v_to-about and _un-_a_rvrs

- Multi-token predicates: map to multiple lemmas

  year- to- year <-> _year+to+year_a_

  according to <-> _according+to_p

- Unknown surface predicates: <unk> (no lemma or POS)

Parsing task

- Asumme tokenization and lemmas given – can be implemented by a pre-processor
- Predict labelled spans and relations
- For now exclude features and scopal edge labels, but framework is flexible

Pierre **Vinken** 61 year old <none> join the board as a <unk> director Nov 29

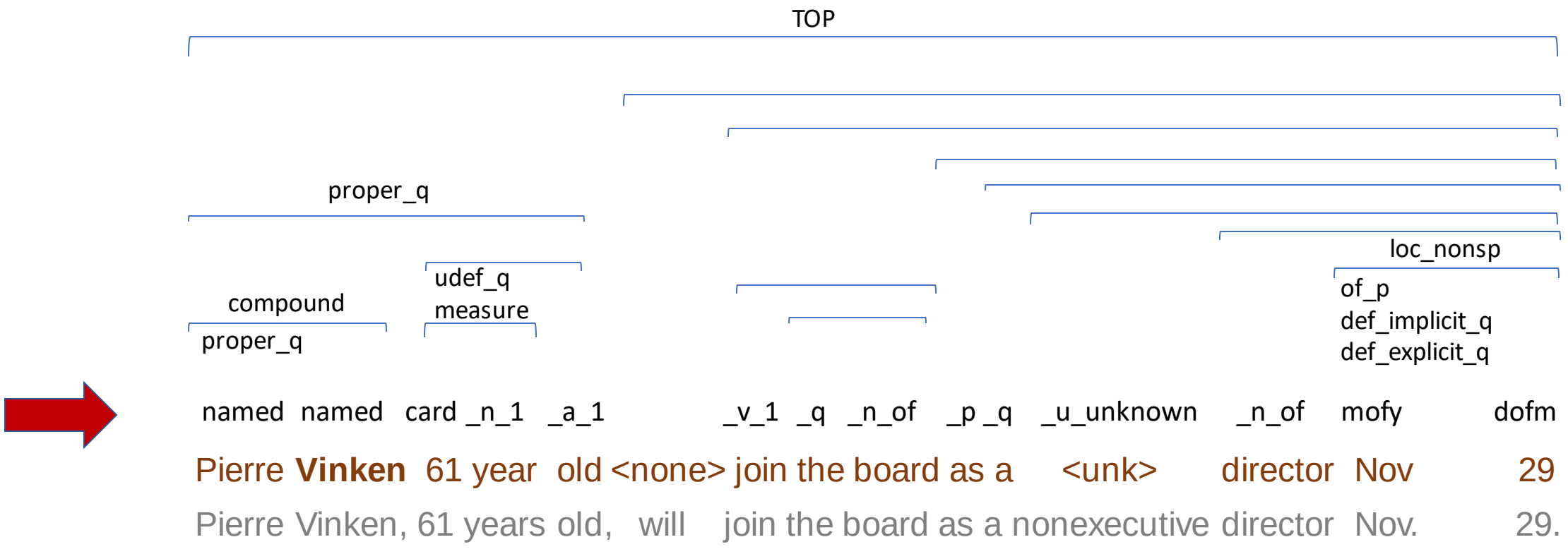
Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

Parsing

- Extend span-based constituency parser to predict:
  - (Unlabelled) syntactic derivation tree
  - Terminal-level semantic labels
  - Non-terminal semantic node label – unless jointly predicted with edges
  - Multiple nodes per span form unary chains

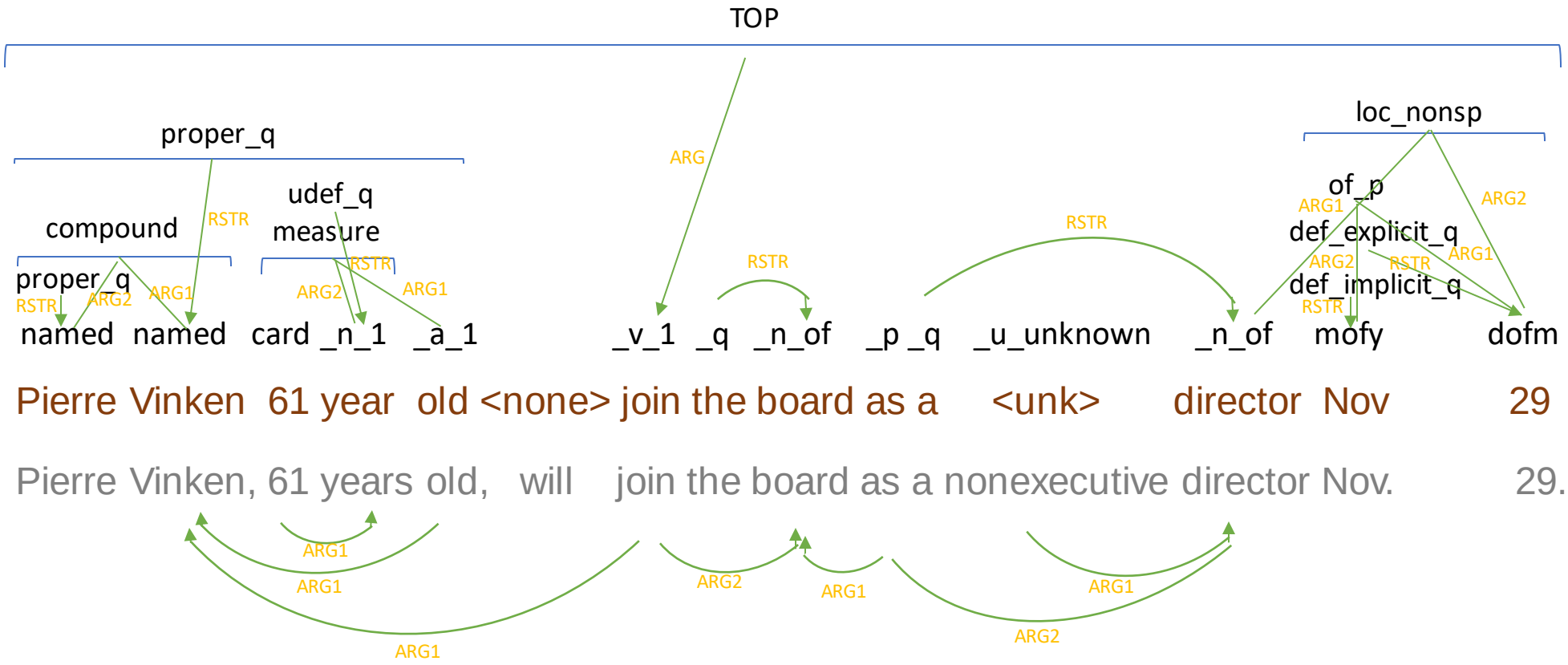
Parsing

- Extended constituency parsing



Parsing

- Extended semantic dependency parser predicts labelled relations (edges) between spans (tree constituents)



Future work

- Implement the parser(s) (!)
- Techniques for joint node (span) and relation prediction

Open questions

- Does this representation make sense linguistically or follow MRS design principles?
- Should some nodes span be modified to fit this framework?
- Should non-terminal semantic nodes be reframed as relations?
- Will incorporating more grammatical information (e.g. SEM-I frames) help? As hard or soft constraints?