Hebrew Dependency Parsing: Initial Results

Yoav Goldberg Michael Elhadad

Ben-Gurion University
of the Negev

IWPT 2009, Paris

Motivation

- We want a Hebrew Dependency parser

Motivation

- We want a Hebrew Dependency parser
- Initial steps:
  - Know Hebrew
  - Create Hebrew Dependency Treebank
  - Experiment with existing state-of-the-art systems

Motivation

- We want a Hebrew Dependency parser
- Initial steps:
  - Know Hebrew
  - Create Hebrew Dependency Treebank
  - Experiment with existing state-of-the-art systems
- Next year:
  - Do better

Motivation

- We want a Hebrew Dependency parser
- **Initial steps:**
  - **Know Hebrew**
  - **Create Hebrew Dependency Treebank**
  - **Experiment with existing state-of-the-art systems**
- Next year:
  - Do better

Know Hebrew

Know Hebrew

- **Relatively free constituent order**
  - Suitable for a dependency based representation

Mostly SVO, but OVS, VSO also possible.
Verbal arguments appear before or after the verb.

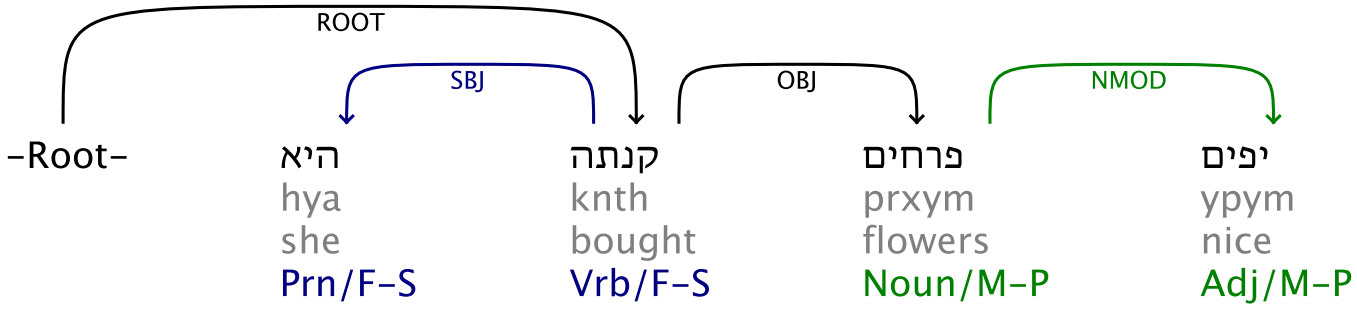
- went from-Israel to-Paris
- went to-Paris from-Israel

...

- to-Paris from-Israel went
- to-Paris went from-Israel

Know Hebrew

- Relatively free constituent order
- **Rich morphology**
  - Many word forms
  - Agreement – noun/adj, verb/subj: should help parsing!



Know Hebrew

- Relatively free constituent order
- Rich morphology
- **Agglutination**
  - Many function words are attached to the next token
  - Together with rich morphology ⇒ Very High Ambiguity
  - **Leaves of tree not known in advance!**

Hebrew Dependency Treebank

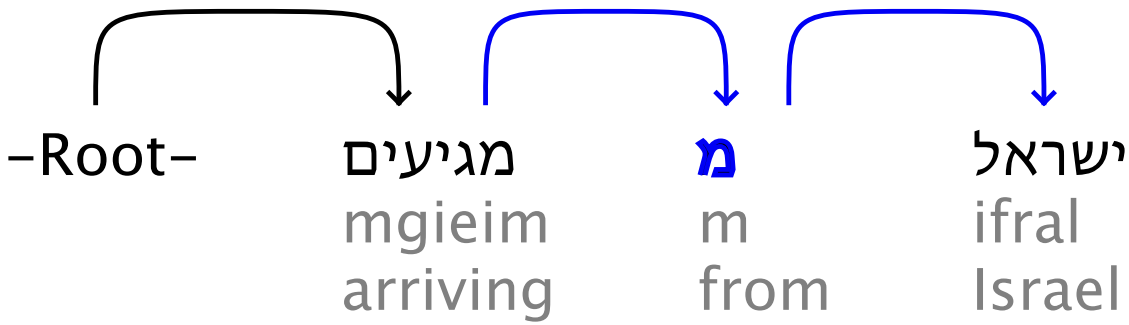
- Converted from Hebrew Constituency Treebank (V2)
  - Some heads marked in Treebank
  - For others: (extended) head percolation table from Reut Tsarfaty
- 6220 sentences
- 34 non-projective sentences

Hebrew Dependency Treebank

- Choice of heads

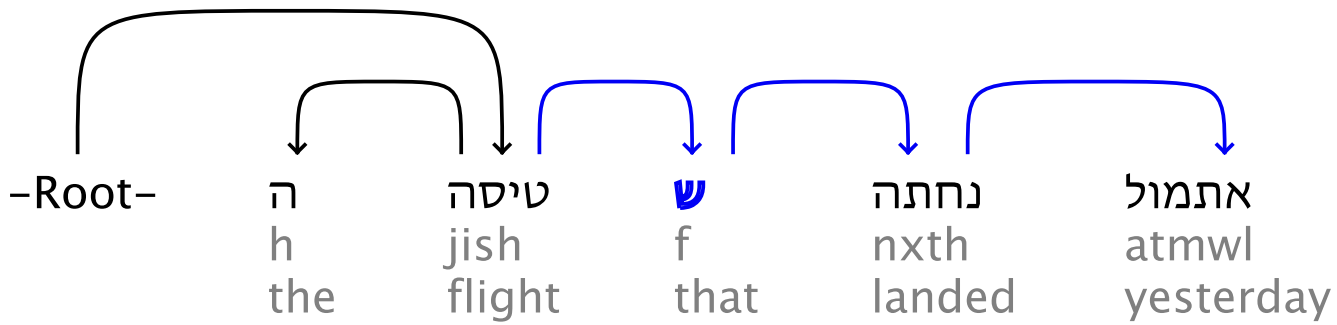
Hebrew Dependency Treebank

- Choice of heads
  - **Prepositions are head of PPs**



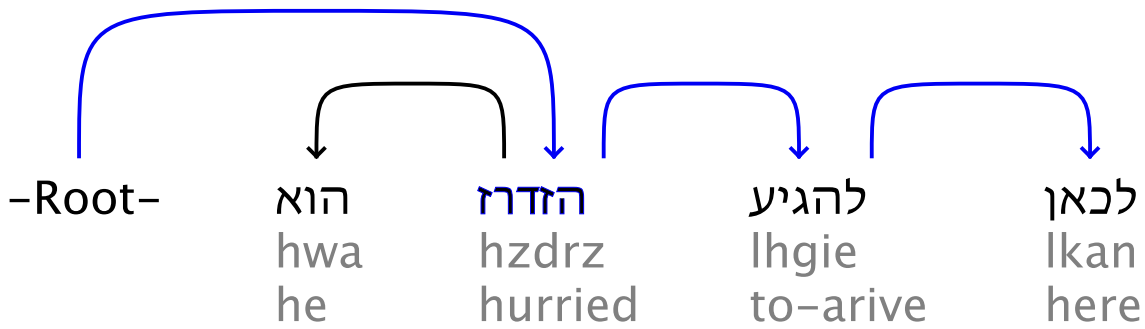
Hebrew Dependency Treebank

- Choice of heads
  - Prepositions are head of PPs
  - **Relativizers are head of Relative clauses**



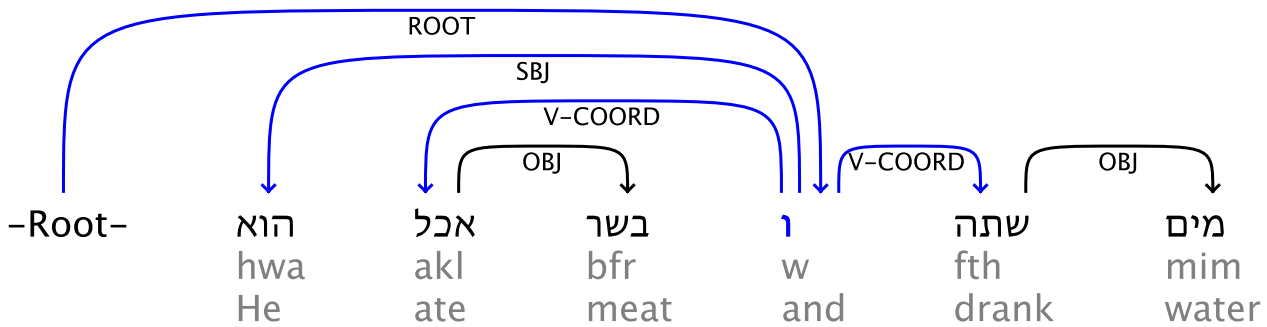
Hebrew Dependency Treebank

- Choice of heads
  - Prepositions are head of PPs
  - Relativizers are head of Relative clauses
  - **Main verb is head of infinitive verb**



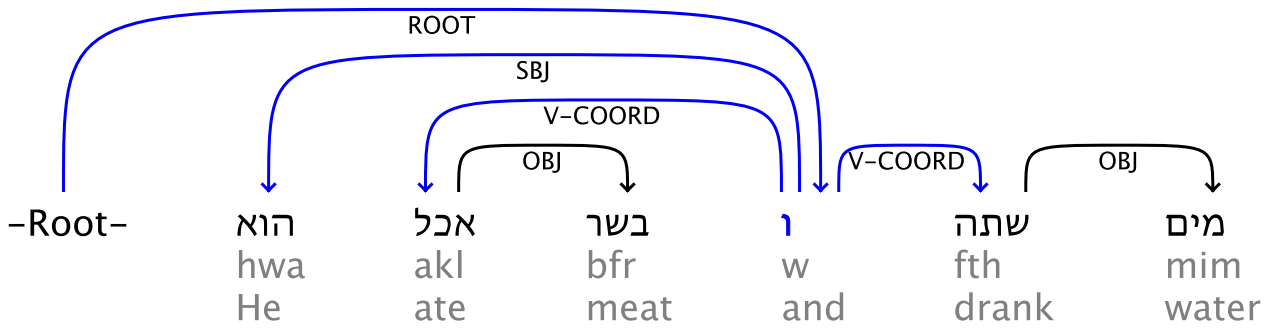
Hebrew Dependency Treebank

- Choice of heads
  - Prepositions are head of PPs
  - Relativizers are head of Relative clauses
  - Main verb is head of infinitive verb
  - **Coordinators are head of Conjunctions**



Hebrew Dependency Treebank

- Choice of heads
  - Prepositions are head of PPs
  - Relativizers are head of Relative clauses
  - Main verb is head of infinitive verb
  - Coordinators are head of Conjunctions ← **hard for parsers**



Hebrew Dependency Treebank

Dependency labels

- Marked in TBv2
  - OBJ
  - SUBJ
  - COMP
- Trivially added
  - ROOT
  - suffix-inflections
- We are investigating ways of adding more labels
- **This work focus on unlabeled dependency parsing.**

Experiments

Parameters

Graph vs. Transitions

How important is lexicalization?

Does morphology help?

Parsers

- Transition based: MALTPARSER (Joakim Nivre)
  - MALT: malt parser, out-of-box feature set
  - MALTARA: malt parser, arabic optimized feature set (should do morphology..)
- Graph based: MST PARSER (Ryan Mcdonald)
  - MST1: first order MST parser
  - MST2: second order MST parser

Experimental Setup

- Oracle setting:
  use gold morphology/tagging/segmentation

- Pipeline setting:
  use tagger based morphology/tagging/segmentation

Results

	Features	MST1	MST2	MALT	MALT-ARA
-MORPH	Full Lex	83.60	84.31	80.77	80.32
	Lex 20	82.99	84.52	79.69	79.40
	Lex 100	82.56	83.12	78.66	78.56
+MORPH	Full Lex	83.60	84.39	80.77	80.73
	Lex 20	83.60	84.77	79.69	79.84
	Lex 100	83.23	83.80	78.66	78.56

Table: **oracle** token segmentation and POS-tagging.

	Features	MST1	MST2	MALT	MALT-ARA
-MORPH	Full Lex	75.64	76.38	73.03	72.94
	Lex 20	75.48	76.41	72.04	71.88
	Lex 100	74.97	75.49	70.93	70.73
+MORPH	Full Lex	73.90	74.62	73.03	73.43
	Lex 20	73.56	74.41	72.04	72.30
	Lex 100	72.90	73.78	70.93	70.97

Table: **Tagger** token segmentation and POS-tagging.

Results

Best oracle result: 84.77%

Best real result: 76.41%

Results

MST2 > MST1 > MALT

Results

MST2 > MST1 > MALT

Simply a better model

Results

MST2 > **MST1** > **MALT**

Partly because of coordination representation

Results

Lexical items appearing > 20 times

$\sim$

all lexical items

Results

With Oracle Morphology

- Morphological features don't really help

Results

With Tagger Morphology

- Morphological features help MALT a little
- Morphological features **hurt** MST a lot

Where do we go from here?

- We have a Hebrew Dependency Treebank
- Realistic performance still too low
- Current models don't utilize morphological information well
  - Can we do better?
- Pipeline model hurt performance
  - Can we do parsing, tagging and segmentation jointly?