НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Dependency grammar and dependency parsing

Francis M. Tyers

ftyers@hse.ru
https://www.hse.ru/org/persons/209454856

Национальный исследовательский университет
«Высшая школа экономики» (Москва)

21 марта 2018 г.

Introduction

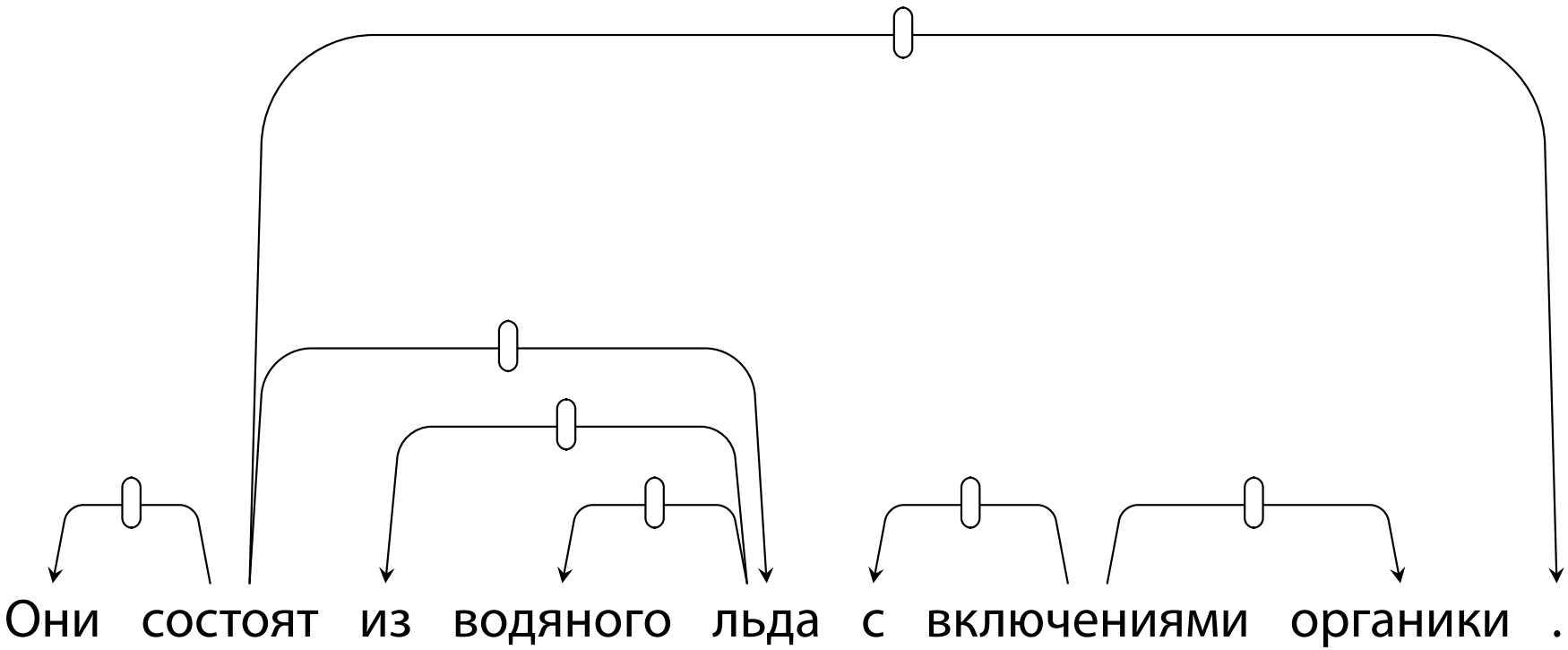
Dependency syntax

- Word based
- No non-terminals
- Words are linked by one-way binary relations
- Relations may be typed or untyped

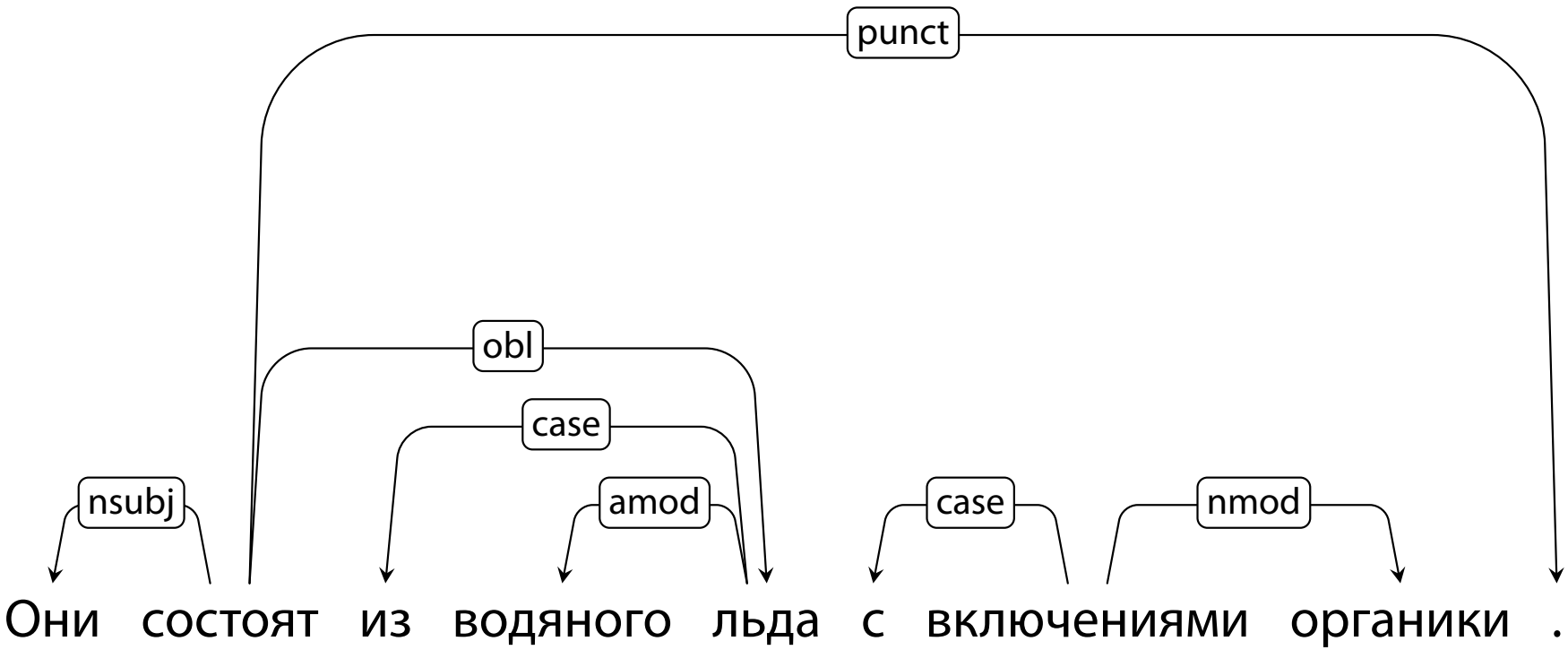
Dependency structure

Они состоят из водяного льда с включениями органики .

Dependency structure



Dependency structure



Terminology

Superior	**Inferior**
Head	Dependent
Governor	Modifier
Regent	Subordinate
Mother	Daughter
Parent	Child

...

Notational variants

x

Notational variants

x

Notational variants

x

Phrase structure

Comparison

Dependency structures explicitly represent:

- head–dependent relations (directed arcs)
- functional categories (arc labels)

Phrase structures explicitly represent:

- phrases (non-terminal nodes)
- structural categories (non-terminal labels)

Heads and dependents

- Criteria for a syntactic relation between a head H and a dependent D in a construction C (Zwicky, 1985)[1]
  1. *H* determines the syntactic category of *C*; *H* can replace *C*
  2. *H* determines the semantic category of *C*; *D* specifies *H*
  3. *H* is obligatory, *D* may be optional
  4. *H* selects *D* and determines optionality of *D*
  5. The form of *D* depends on *H* (agreement or government)
  6. Linear position of *D* is specified with reference to *H*
- An issue:
  - Syntactic (and morphological) versus semantic criteria

[1] Zwicky, A. (1985) "Heads" *Journal of Linguistics*, 21:1–29

Some fuzzy cases

- Complex verb groups (auxiliary–main verb)
- Subordinate clauses (complementiser–verb)
- Coordination (coordinator–conjuncts)
- Adpositional phrases (adposition–nominal)
- Punctuation

Dependency graphs

A dependency graph, G

- a set of V nodes,
- a set of A arcs,
- a linear precedence order $<$ on V

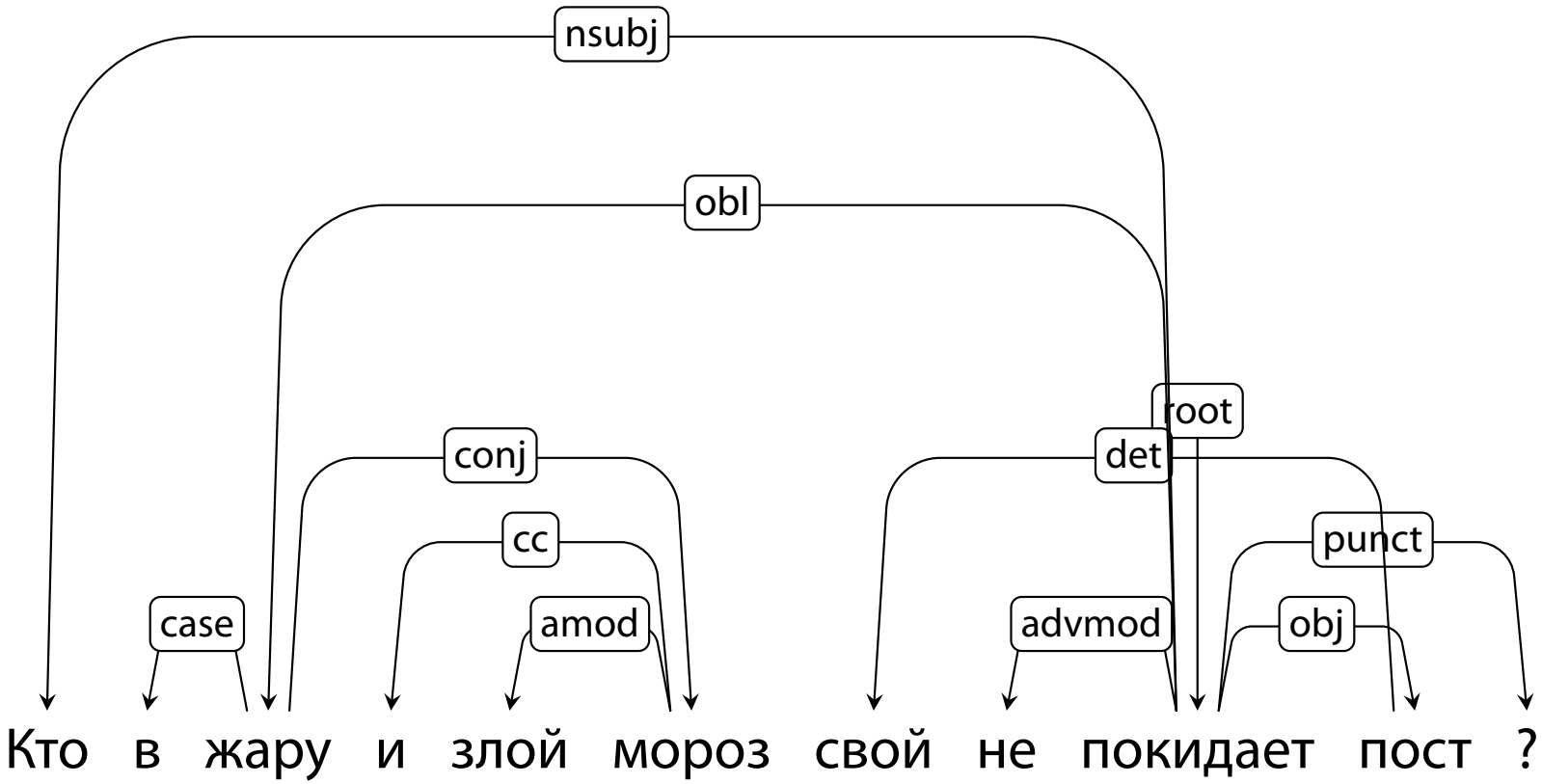
Labelled graphs:

- Nodes in V are labelled with word forms (and annotation)
- Arcs in A are labelled with dependency types

...

Conditions

Projectivity



Parsing methods

Transition-based

General idea

Components

Data structures:

- Stack:
  - Starts as containing only the ROOT
- Buffer
  - Starts as containing the full sentence
- Arcs
  - Starts as empty

Operations:

- SHIFT: Take the word on top of the buffer and put it on the stack
- LEFT-ARC: Make the word at the top of the stack the head of the word below it
  - Then remove the word at the top
- RIGHT-ARC: Make the word second from top the head of the word above it
  - Then remove the second from top word

Example

ROOT Мы пошли домой

Stack	**Buffer**
ROOT	Мы пошли домой

Example

SHIFT

ROOT Мы пошли домой

Stack	**Buffer**
ROOT Мы	пошли домой

Example

SHIFT

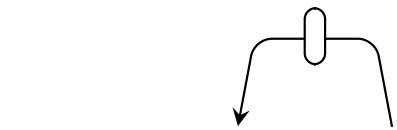
ROOT Мы пошли домой

Stack **Buffer**

ROOT Мы пошли домой

Example

LEFT-ARC

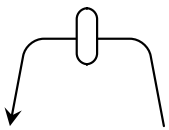


ROOT Мы пошли домой

Stack	**Buffer**
ROOT пошли	домой

Example

SHIFT



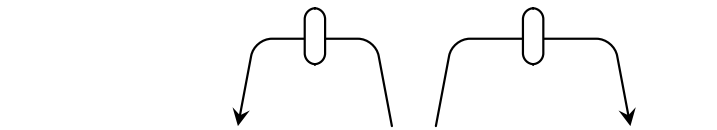
ROOT Мы пошли домой

Stack **Buffer**

ROOT пошли домой

Example

RIGHT-ARC

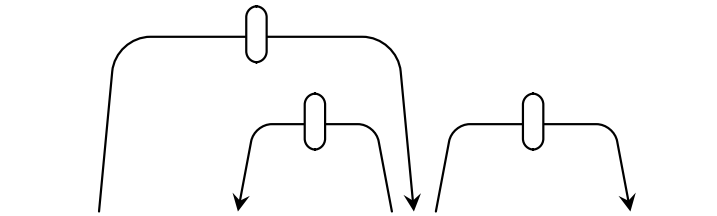


Stack **Buffer**

ROOT пошли

Example

RIGHT-ARC



Stack **Buffer**

ROOT

Configurations

A **configuration** is a snapshot of the state of the parser at a given time.

- A stack: Representing the word(s) currently being processed
- A buffer: Representing the remaining words
- A set of arcs representing a (partial) tree

We can conceive parsing as transitioning from one configuration to another via an operation.

Oracle

How do we get the sequence of operations?

Deterministic algorithm:

- LEFT-ARC: Configuration has arc from the top of stack to the word below
- RIGHT-ARC: Configuration has arc from the of the stack to the first word in the input buffer
  - In addition: The dependent must have no dependents of its own
- SHIFT: All other cases

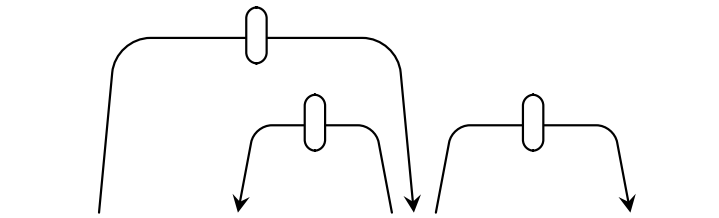
Oracle

How do we get the sequence of operations?

Deterministic algorithm:

- LEFT-ARC: Configuration has arc from the top of stack to the word below
- RIGHT-ARC: Configuration has arc from the of the stack to the **first word in the input buffer**
  - In addition: The **dependent** must have no dependents of its own
- SHIFT: All other cases

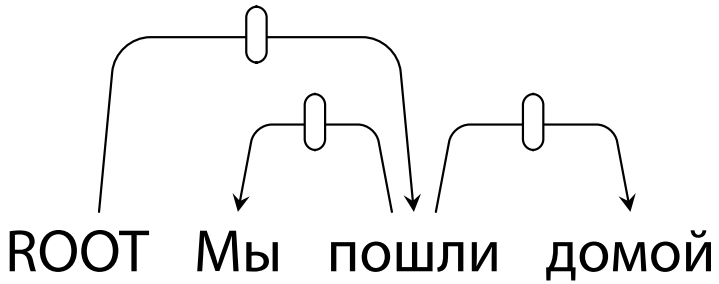
Parsing with an oracle



Stack	**Buffer**
ROOT	Мы пошли домой

- Is there an arc from the first word in the buffer to the top of the stack?
  - (Мы, ROOT)
- Is there an arc from the top of the stack to the first word in the buffer?
  - (ROOT, Мы)
- $\rightarrow$ then SHIFT

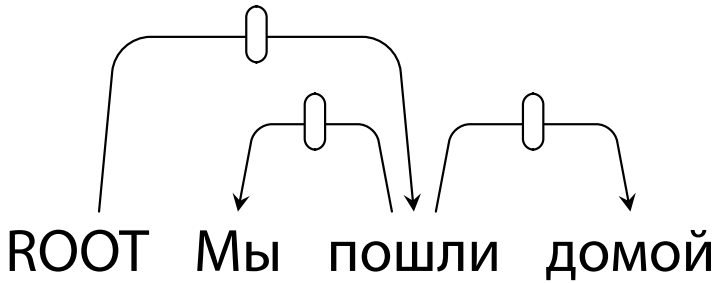
Parsing with an oracle



Stack	Buffer
ROOT Мы	пошли домой

- Is there an arc from the first word in the buffer to the top of the stack?
  - (пошли, Мы) — YES, LEFT-ARC
- Is there an arc from the top of the stack to the first word in the buffer?
  - (Мы, пошли)

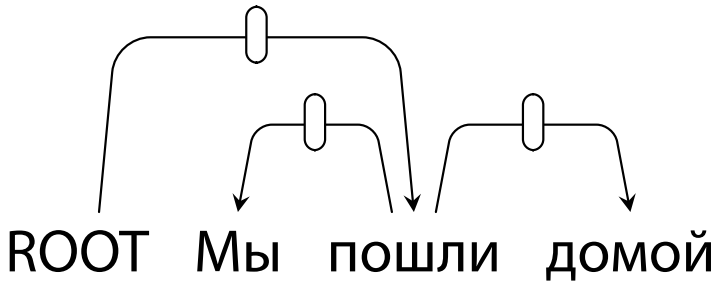
Parsing with an oracle



Stack	Buffer
ROOT	пошли домой

- Is there an arc from the first word in the buffer to the top of the stack?
  - (пошли, ROOT)
- Is there an arc from the top of the stack to the first word in the buffer?
  - (ROOT, пошли) – YES, but *пошли* still has dependents
- $\rightarrow$ then SHIFT

Parsing with an oracle



Stack	Buffer
ROOT пошли	домой

- Is there an arc from the first word in the buffer to the top of the stack?
  - (домой, пошли)
- Is there an arc from the top of the stack to the first word in the buffer?
  - (пошли, домой) – YES, and *домой* has no dependents
  - $\rightarrow$ RIGHT-ARC

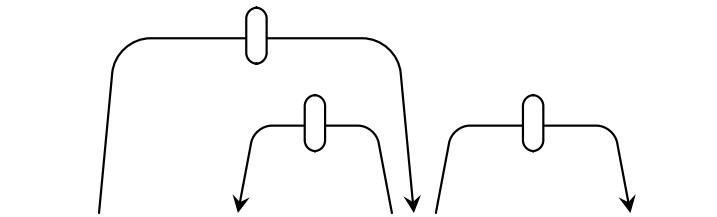
Training data

The “only” training data required is a treebank.

- Collection of sentences annotated for dependency structure
- Universal dependencies: 67 languages, 100s of treebanks

Data trains a classifier to predict a transition from a configuration.

Features



Stack	Buffer
ROOT пошли	домой

Features indexed by address (in stack or buffer) and attribute name.

Traditional:

- (Stack[0], Form) = пошли
- (Buffer[0], Form) = домой
- (Stack[0], UPOS) = VERB
- (Stack[1], Form) = ROOT

Can return NULL.

Indicator:

- Combinations of such features, e.g.
  - (Stack[0], Form) = пошли & (Buffer[0], UPOS) = ADP

Features begone!

Extensions

Graph-based

Basic model

Compared to transition based

Chu-Liu-Edmonds

Evaluation

Simple evaluation:

- Unlabelled attachment score, **UAS**: correct heads/total heads
- Labelled attachment score, **LAS**: (correct heads+labels)/total heads

Shared tasks

CoNLL 2018

Ace the exam!