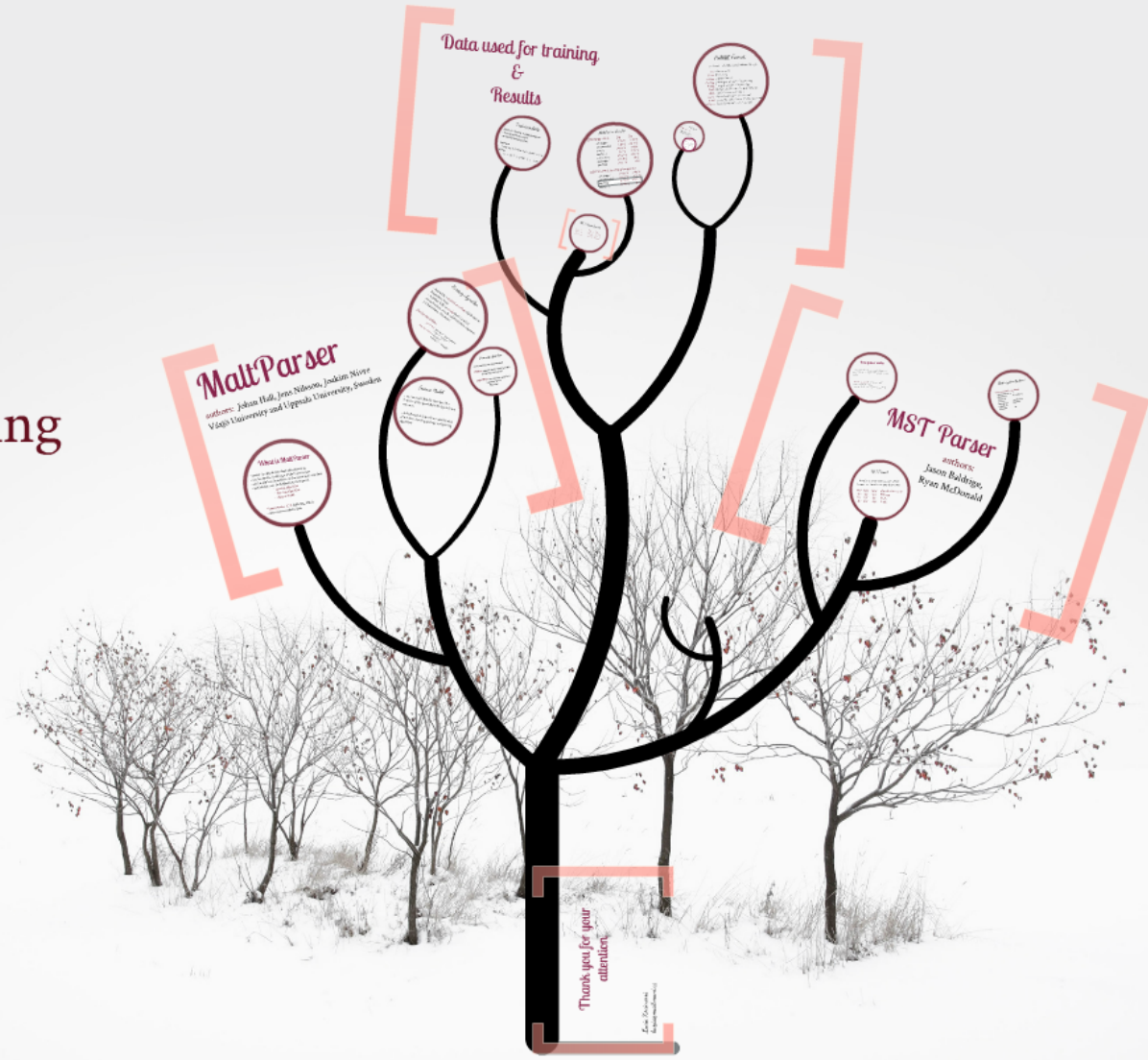
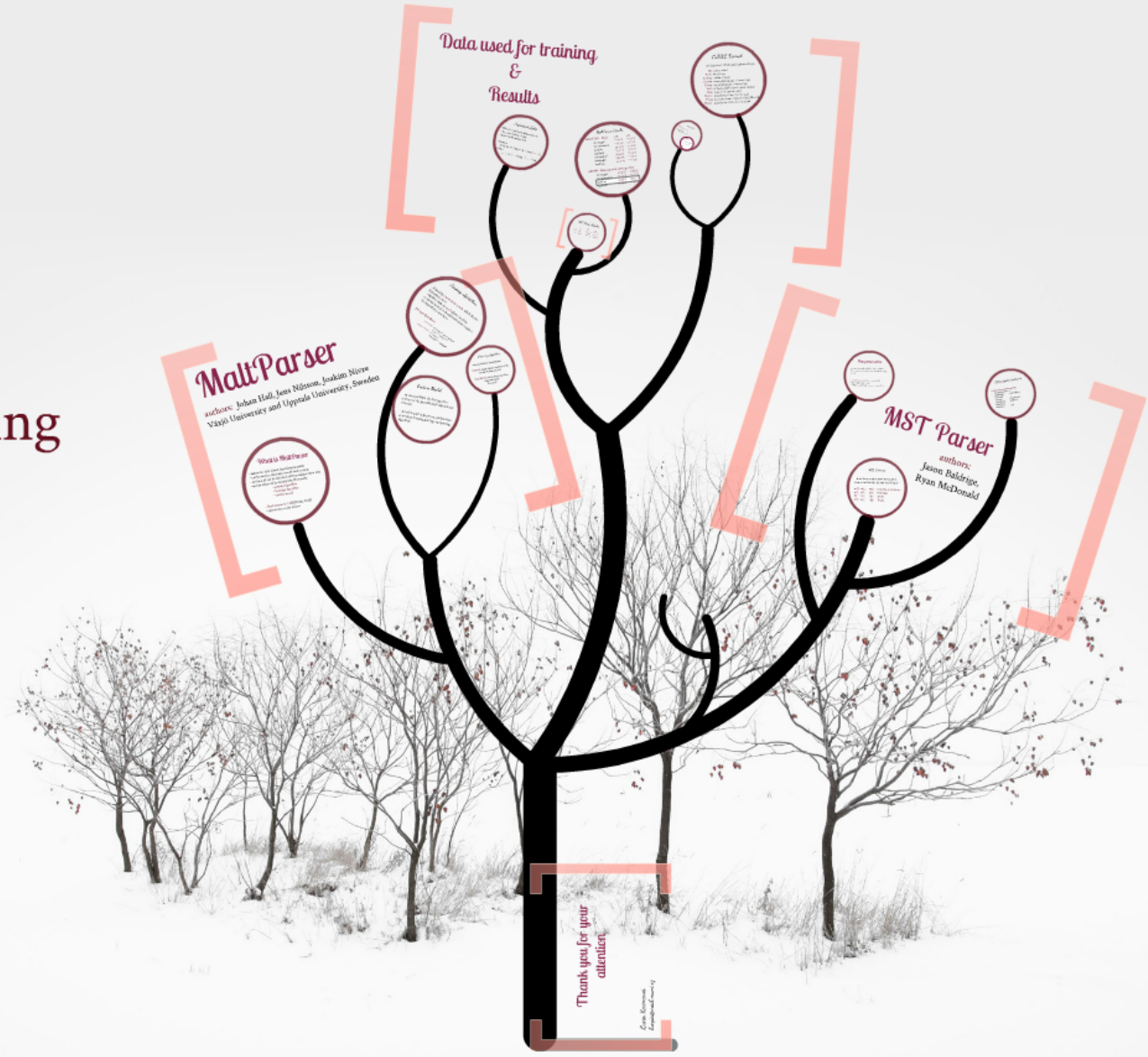
Trees

or what is left after parsing



Trees

or what is left after parsing



MaltParser

authors: Johan Hall, Jens Nilsson, Joakim Nivre
Växjö University and Uppsala University, Sweden

What is MaltParser

- system for data driven dependency parsing
- can be used for inducing a model from corpus
- such model can be used for further parsing of new data
- specification can be divided into three parts:
  - parsing algorithm
  - learning algorithm
  - feature model

- latest version 1.7.2 (25th Sep. 2012)
- open source coded in Java

Parsing algorithm

- defined by a transition system which derives dependency trees together with an oracle that is used for reconstruction of each valid transition sequence for dependency structures

Parsing algorithms:

projective: nivreeager*, nivrestandard*, covproj*, stackproj*+

non-projective: covnonproj*, stackeager*+, stacklazy*+

Feature Model

- an external XML file that specifies features of the partially built dependency structure

- default model: depends on combination of machine learning package and parsing algorithm

Learning algorithm

- two machine learning packages:

LIBSVM: support vector machines with kernels (Chang and Lin)

LIBLINEAR: various linear classifiers including SVMs (Fan et al.)

What is MaltParser

- system for data driven dependency parsing
- can be used for inducing a model from corpus
- such model can be used for further parsing of new data
- specification can be divided into three parts:
  - ~ parsing algorithm
  - ~ learning algorithm
  - ~ feature model

- latest version 1.7.2 (25th Sep. 2012)
- open source coded in Java

Parsing algorithm

- defined by a transition system which derives dependency trees
- together with an oracle that is used for reconstruction of each valid transition sequence for dependency structures

Parsing algorithms:

projective:	nivreeager*, nivrestandard*, covproj^, stackproj**
non-projective:	covnonproj^, stackeager**, stacklazy**

Learning algorithm

- two machine learning packages:

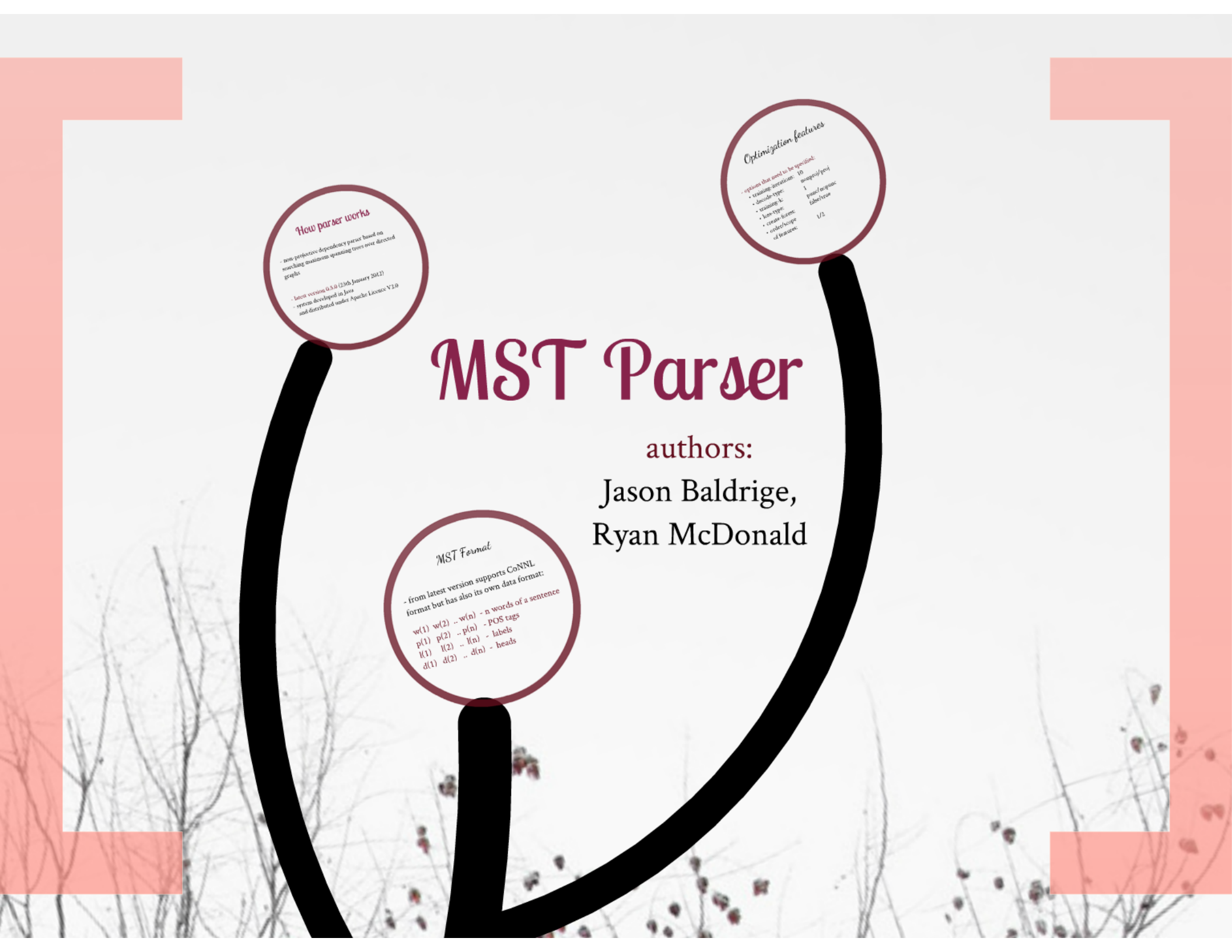
LIBSVM: support vector machines with kernels (Chang and Lin)

LIBLINEAR: various linear classifiers including SVMs (Fan et al.)

Feature Model

- an external XML file that specifies features of the partially built dependency structure

- default model: depends on combination of machine learning package and parsing algorithm


How parser works
- non-projective dependency parser based on searching maximum spanning trees over directed graphs
- latest version 0.3.0 (23th January 2012)
- system developed in Java
and distributed under Apache Licence V2.0
Optimization features
- options that need to be specified:
• training-iterations: 10
• decode-type: nonproj/proj
• training-k: 1
• loss-type: punct/nopunc
• create-forest: false/true
• order/scope of features: 1/2
MST Format
- from latest version supports CoNNL format but has also its own data format:
w(1) w(2) .. w(n) - n words of a sentence
p(1) p(2) .. p(n) - POS tags
l(1) l(2) .. l(n) - labels
d(1) d(2) .. d(n) - heads
MST Parser
authors:
Jason Baldrige,
Ryan McDonald

How parser works

- non-projective dependency parser based on searching maximum spanning trees over directed graphs

- latest version 0.5.0 (23th January 2012)
- system developed in Java
  and distributed under Apache Licence V2.0

Optimization features

- options that need to be specified:
  - training-iterations: 10
  - decode-type: nonproj/proj
  - training-k: 1
  - loss-type: punc/nopunc
  - create-forest: false/true
  - order/scope of features: 1/2

MST Format

- from latest version supports CoNNL format but has also its own data format:

w(1) w(2) .. w(n) - n words of a sentence
p(1) p(2) .. p(n) - POS tags
l(1) l(2) .. l(n) - labels
d(1) d(2) .. d(n) - heads

Data used for training & Results

Training data

- PDT 2.0 used for training purposes
- data anotated on a-layer
- manually disambiguated

sentences

train: 68 495 | dtest: 9 270 | etest: 10 148

tokens

1 171 191 | 158 962 | 173 586

CoNNL Format

- introduced in CoNNL shared task workshops

ID	token counter
Form	word form
Lemma	lemma of word
Cpostag	coarse-grained part-of-speech tag
Postag	fine-grained part-of speech tag
Feats	syntactic and/or morfological features
Head	head of the current token
Deprel	dependency relation to the head
Phead	projective head of current token (ID or 0)
Pdeprel	dependency relation to the phead

MaltParser Results

LIBLINEAR ~ 2hod.	UA	LA
nivreeager	79.99 %	71.89 %
nivrestandard	71.43 %	64.73 %
covproj	80.13 %	71.43 %
stackproj	79.67 %	73.99 %
covnonproj	80.58 %	74.95 %
stackeager	82.54 %	77.14 %
stacklazy	83.17 %	77.74 %

LIBSVM ~20hod. (použitý splitting trick)		
nivreeager	83.21 %	78.42 %
nivrestandard	81.51 %	76.37 %
stacklazy	85.02 %	80.05 %
stackproj	83.00 %	77.47 %

MST Parser Results

	LA	UA
opt. 1 ~ 5 h.	69.19 %	77.73 %
opt. 2 ~ 4 h.	75.34 %	83.01 %
opt. 3 ~ 4.5 h.	75.39 %	83.04 %

Evaluation

CoNNL shared Task results:

unlabeled accuracy ~ 87.30 %

labeled accuracy ~ 80.38

Evaluation script

CoNNL Shared tasks used various scripts, eg. Randomized Parsing Evaluation Comparator

own script developed

Thank you for your attention

Lucia Kocincová
lucyia@mail.muni.cz