Rudolf Rosa
rosa@ufal.mff.cuni.cz

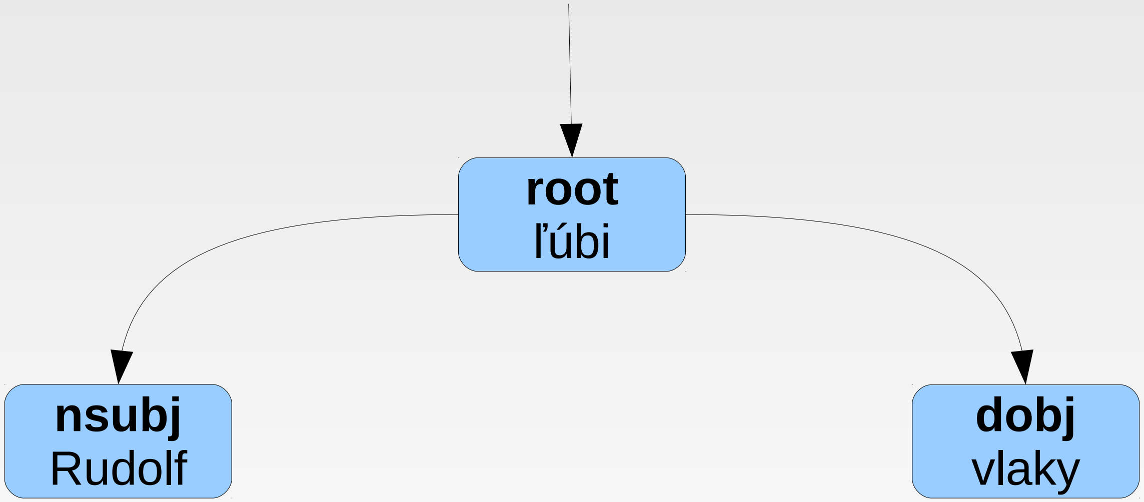
Cross-lingual Transfer of Dependency Parsers

Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Monday seminar, ÚFAL MFF UK, Praha, 11 Dec 2017

The problem of parsing

- input: text in a *target* language, e.g. Slovak:
  - *Rudolf ľúbi vlaky (“Rudolf likes trains”)*
- output: syntactic analysis of the text (UD tree)



A solution

- if we **have** a target treebank
  - train a parser on the target treebank (UDPipe)
  - apply the parser to the text, obtain a parse tree

tagger&parser

A solution

tagger&parser

- if we **have** a target treebank
  - train a parser on the target treebank (UDPipe)
  - apply the parser to the text, obtain a parse tree
- if we **don't have** a target treebank
  - take a treebank for a *source* language (e.g. Czech)
  - translate it into the *target* language (MT, e.g. Moses)
    - conversion to the previous case
  - train a parser on the *pseudo-target* treebank
  - apply the parser to the text, obtain a parse tree
  - (or: annotate some data in the target language)

A solution

~ 70 languages, news/books/wiki

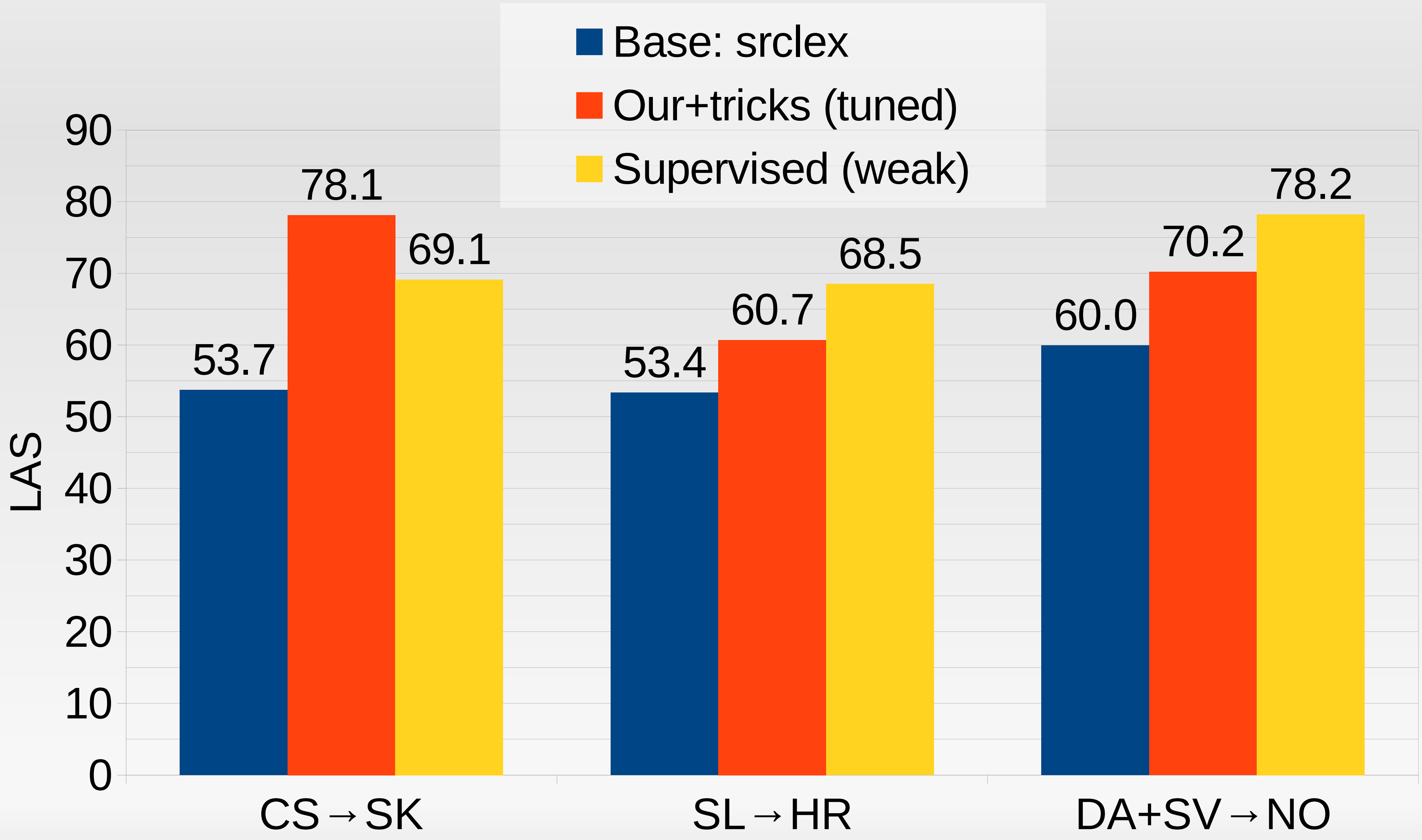
tagger&parser

- if we **have** a target treebank
  - train a parser on the target treebank (UDPipe)
  - apply the parser to the text, obtain a parse tree
- if we **don't have** a target treebank

~ 7000 languages

  - take a treebank for a *source* language (e.g. Czech)
  - translate it into the *target* language (MT, e.g. Moses)
    - conversion to the previous case
  - train a parser on the *pseudo-target* treebank
  - apply the parser to the text, obtain a parse tree
  - (or: annotate some data in the target language)

An evaluation (Rosa+, 2017)



Outline

<u>Cross-lingual Transfer of Dependency Parsers</u>

- Brief overview of the problem and a solution
- Why and how we parse text
- Without Machine Translation: Delex parsing
- How to do Machine Translation
- How to choose the source language
- How to combine multiple sources

Outline

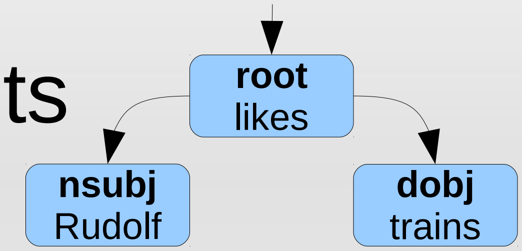
Cross-lingual Transfer of Dependency Parsers

- Brief overview of the problem and a solution
- **Why and how we parse text**
- Without Machine Translation: Delex parsing
- How to do Machine Translation
- How to choose the source language
- How to combine multiple sources

Why to parse text

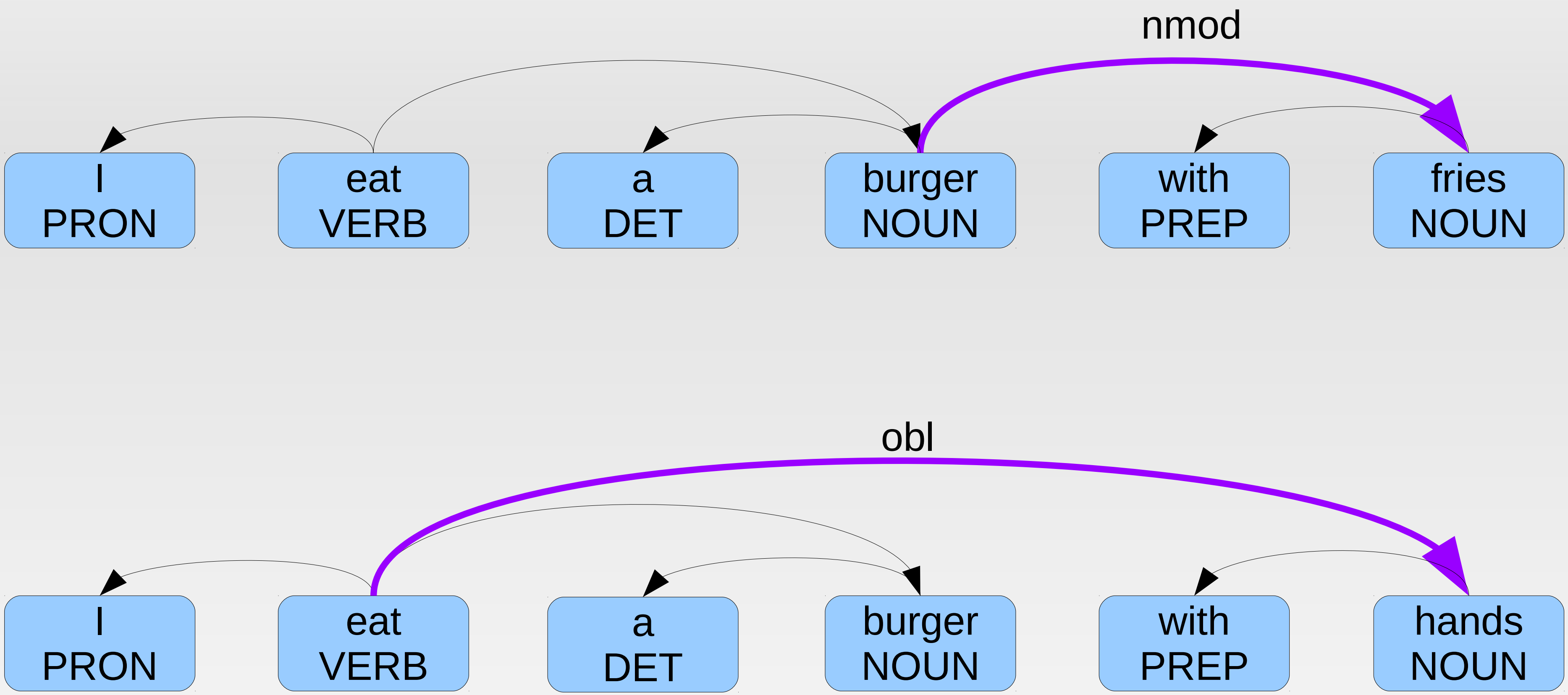
- to understand its structure (→ and its meaning)
- in formal linguistics
  - automatic pre-analysis for corpus linguistics
- in computational linguistics
  - traditionally: preprocessing of input for further tasks
  - modern way: train end2end NN on labelled text data
  - insufficient data for the end task: anything can help
    - parsing as an abstraction over the input
    - rules/heuristics to solve the task
    - e.g. Depfix, coreference, chatbot, text generation...

How does a parser work

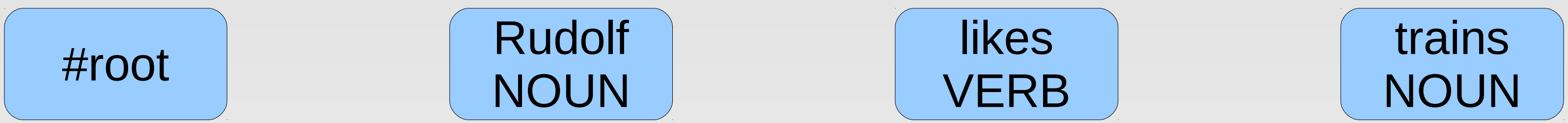


- ML task: for each word, determine its head word and the relation to it
  - dependency trees vs. phrase-structure trees
- input representation features – on dependent, its potential head, as well as context words:
  - word distance (shorter edges more likely)
  - word order (left/right branching)
  - part-of-speech tags – the killer feature (±morphofeats)
  - word forms – the disambiguation feature
- inference algorithm: e.g. MST or shift-arc parsing

Lexicalization for disambiguation

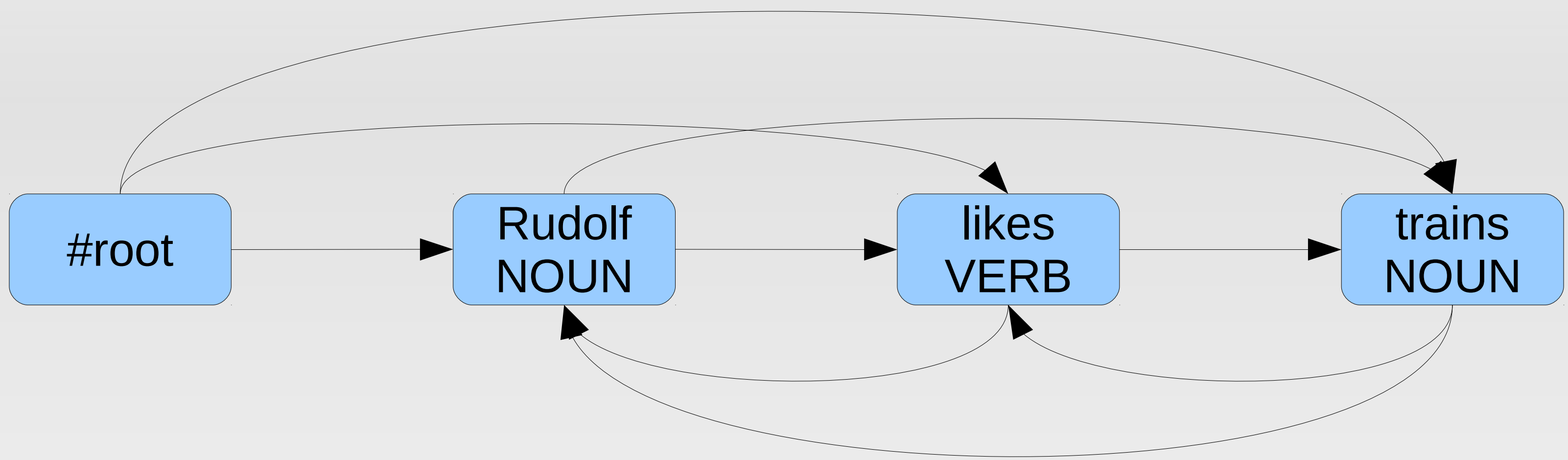


Maximum Spanning Tree Parser



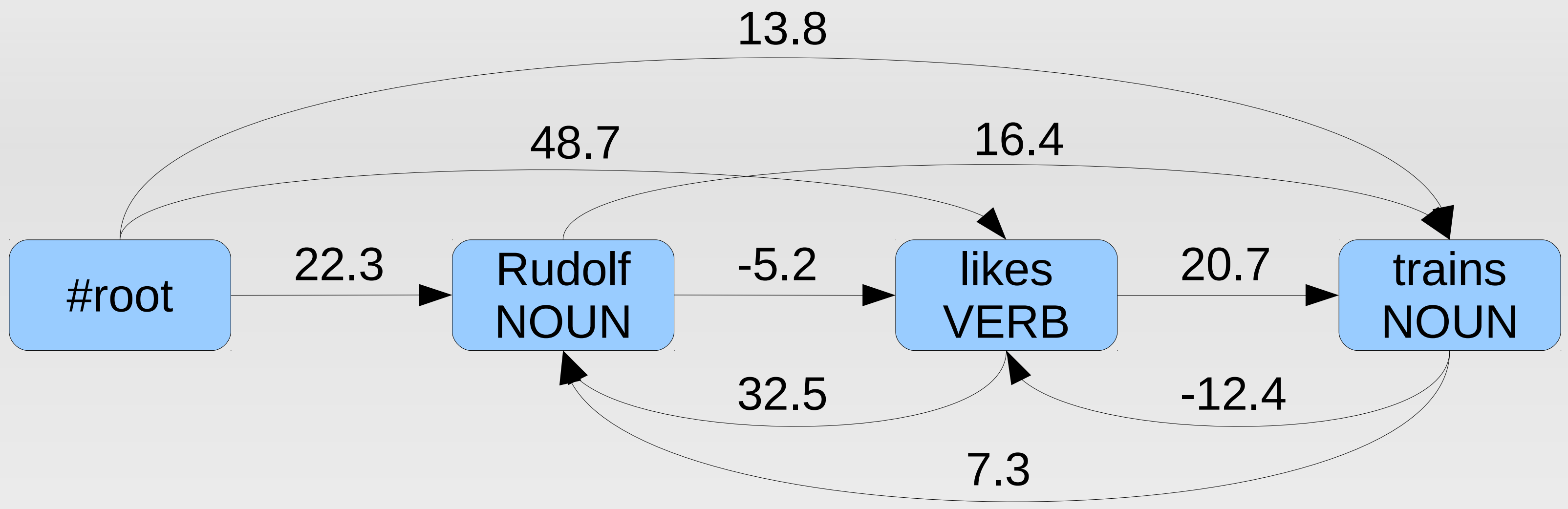
- graph
- words → nodes + virtual root node

Maximum Spanning Tree Parser



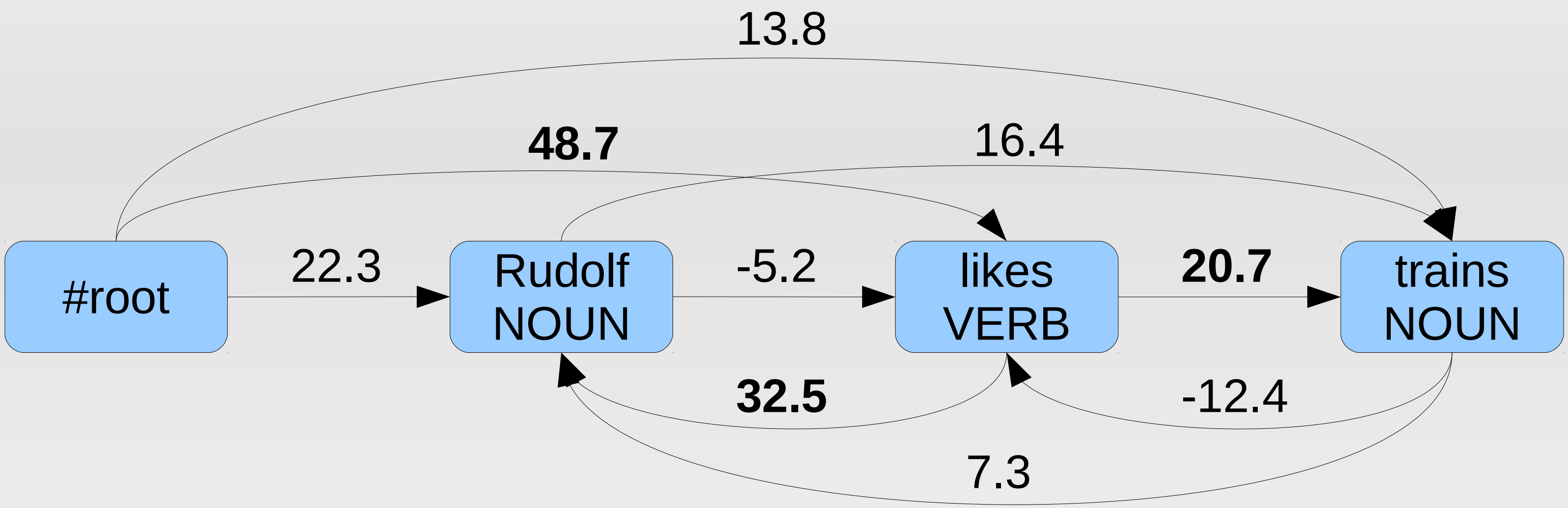
- nearly-complete directed graph
  - all possible dependency edges

Maximum Spanning Tree Parser



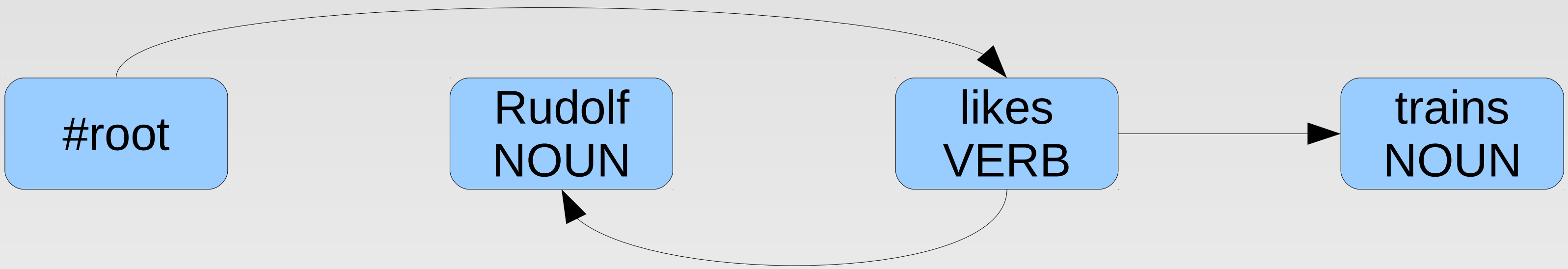
- weighted graph
- edge weight = sum of weights of features active on that edge (weights come from trained model)

Maximum Spanning Tree Parser



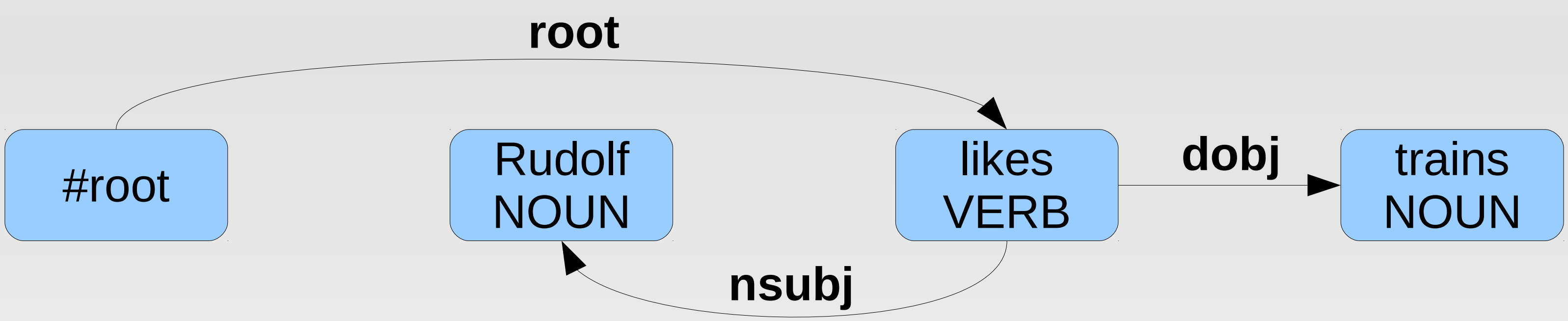
- MST algorithm: Chu-Liu-Edmonds or Eisner

Maximum Spanning Tree Parser



- unlabelled parse tree

Maximum Spanning Tree Parser



- labelling: a Markov chain labeller

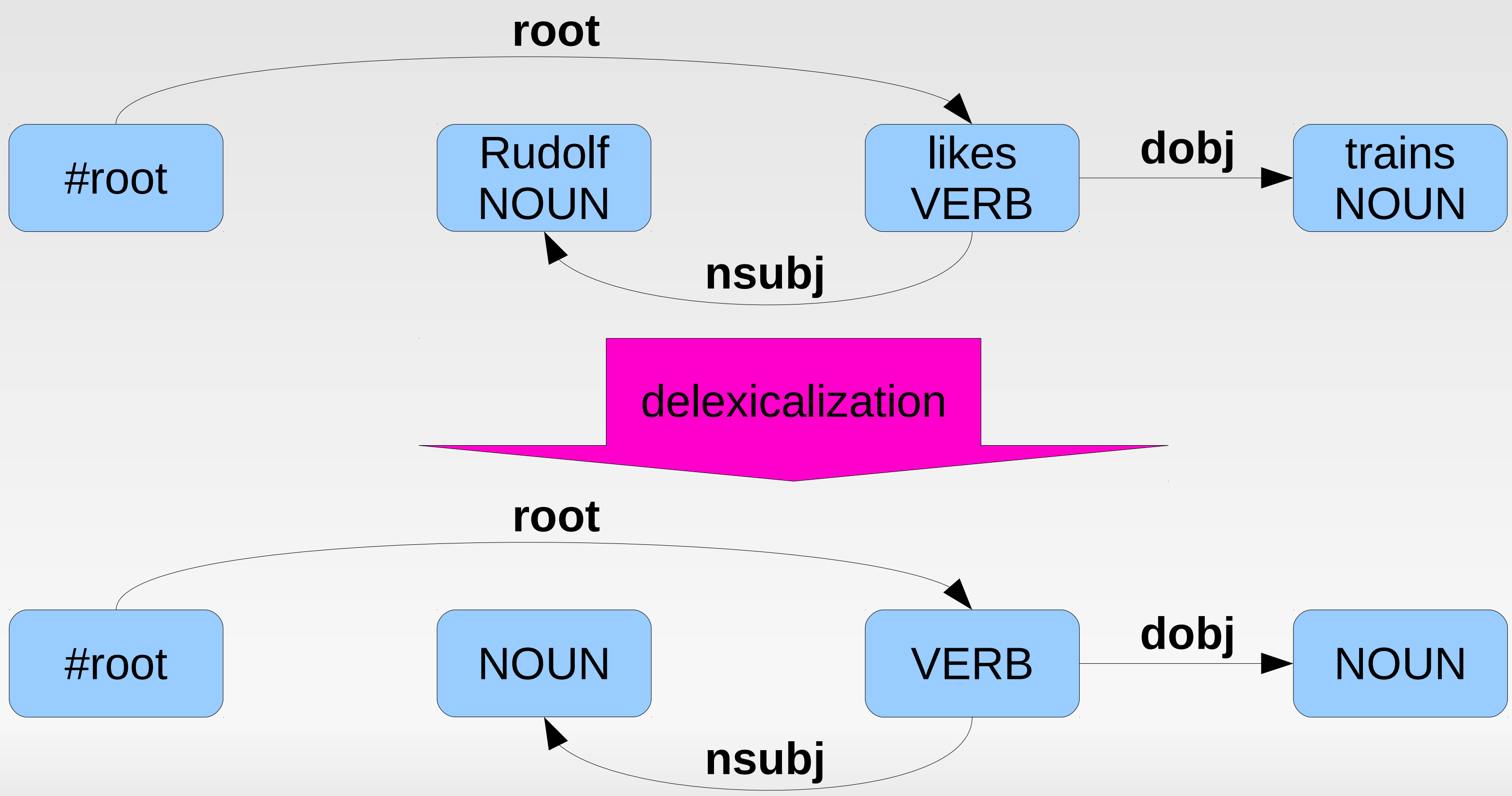
Outline

Cross-lingual Transfer of Dependency Parsers

- Brief overview of the problem and a solution
- Why and how we parse text
- **Without Machine Translation: Delex parsing**
- How to do Machine Translation
- How to choose the source language
- How to combine multiple sources

Delexicalized parsing

- delex parsing = without lexical features
  - delete word forms from data, use POS & position



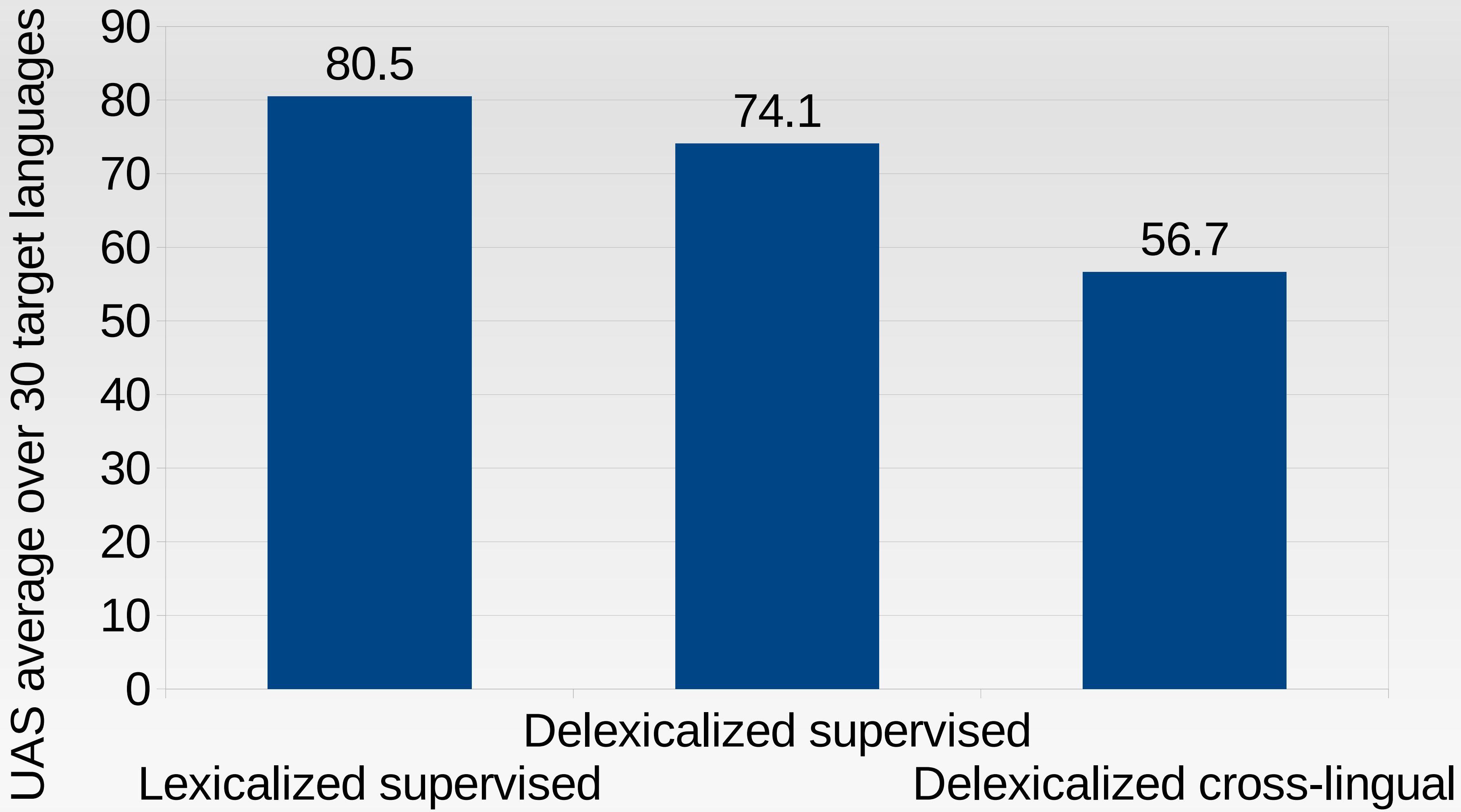
Delexicalized parsing: Motivation

- POS tags = the killer feature
  - supervised mono: delex ~70%, lex ~80%
- universal POS tags shared across languages
  - no need for translation
  - a delex parser is a “universal” parser
  - easy combination of multiple source languages
- simple task, easy to experiment with
  - all early work on cross-lingual parsing uses delex

Delex parsing: Harmonization

- source and target must use the same annotation
  - harmonization of existing treebanks/new annotation
- HamleDT (ÚFAL) ← PDT & Interset (existing data)
- uni-dep-tb (Google) ← Stanford Deps (new data)
- Universal Dependencies, now v2.1 (existing + new)
  - 17 universal POS ← Univ. POS (Petrov+, 2011)
  - 21 universal features ← Interset (Zeman, 2008)
  - 37 universal dependencies ← USD (de Marneffe+, 2014)
  - still some heterogeneity – worth addressing...

Delexicalized parsing: Evaluation



Delexicalized parsing: Problems I.

- assumes having a tagger for target language
  - focus: under-resourced languages
    - typically no tagger available
    - has tagger → often also has treebank
  - cross-lingual tagger projection needs parallel texts
    - why not also use those for MT-based lexicalization?
    - lexicalized parsing usually better than delexicalized
    - maybe different in case of small parallel data?
      - Bible paper (Agić+, 2015) and further papers

Delexicalized parsing: Problems II.

- assumes strong source-target grammar similarity
  - true for all cross-lingual methods
  - but lexical information can help to disambiguate!

a	red	strawberry	and	a	yellow	banana
DET	ADJ	NOUN	SCONJ	DET	ADJ	NOUN

una	fragola	rossa	e	una	banana	gialla
DET	NOUN	ADJ	SCONJ	DET	NOUN	ADJ

  - more sensitive to choice of source language
    - word order, auxiliaries, morphology, data size...
    - wait till end of talk!

Outline

Cross-lingual Transfer of Dependency Parsers

- Brief overview of the problem and a solution
- Why and how we parse text
- Without Machine Translation: Delex parsing
- **How to do Machine Translation**
- How to choose the source language
- How to combine multiple sources

What to translate

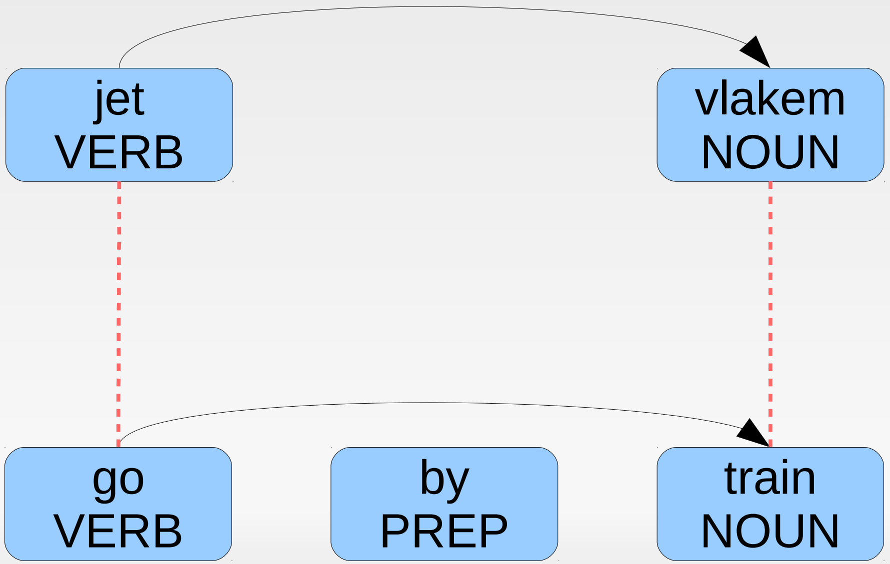
- translate input text (target → source)
  - use a ± standard source parser to parse it
  - …translation done at inference
- translate training treebank (source → target)
  - train a pseudo-target parser on the translated TB
  - …translation done at training
- other options
  - parse source side of parallel text, project trees
  - translate the word forms in the trained model
  - …

What to translate

- translate input text (target→source)
- **translate training treebank (source→target)**
  - empirically better results
  - parser trained on noisy data→hopefully more robust
  - can employ monolingual target texts
    - MT: train a target language model
    - parser: pre-train word embeddings (NN parser)
  - easier combination of multiple sources
  - simpler inference – can directly parse target texts

How to translate

- source and target sentences do not map 1:1
  - problems even with very similar languages
  - obviously worse for more distant languages



Solutions to non-isomorphism

- ignore it, act if the languages align 1:1
  - super-simple – Moses with phrase length = 1
    - ± reordering, ± N:N alignment (e.g. 2:2)
  - lower-quality MT, but seems not that crucial
- complex projection heuristics
  - can use M:N word-alignment and phrase-based MT
    - or even NMT, but maybe that's an overkill
  - omit some nodes, guess some edges&deprels...
  - MT less noisy x projection more noisy
  - seems similar for close langs, better for distant langs

that's what I do now

Hwa+ (2005), Ramasamy (2014), Tiedemann+ (2014)

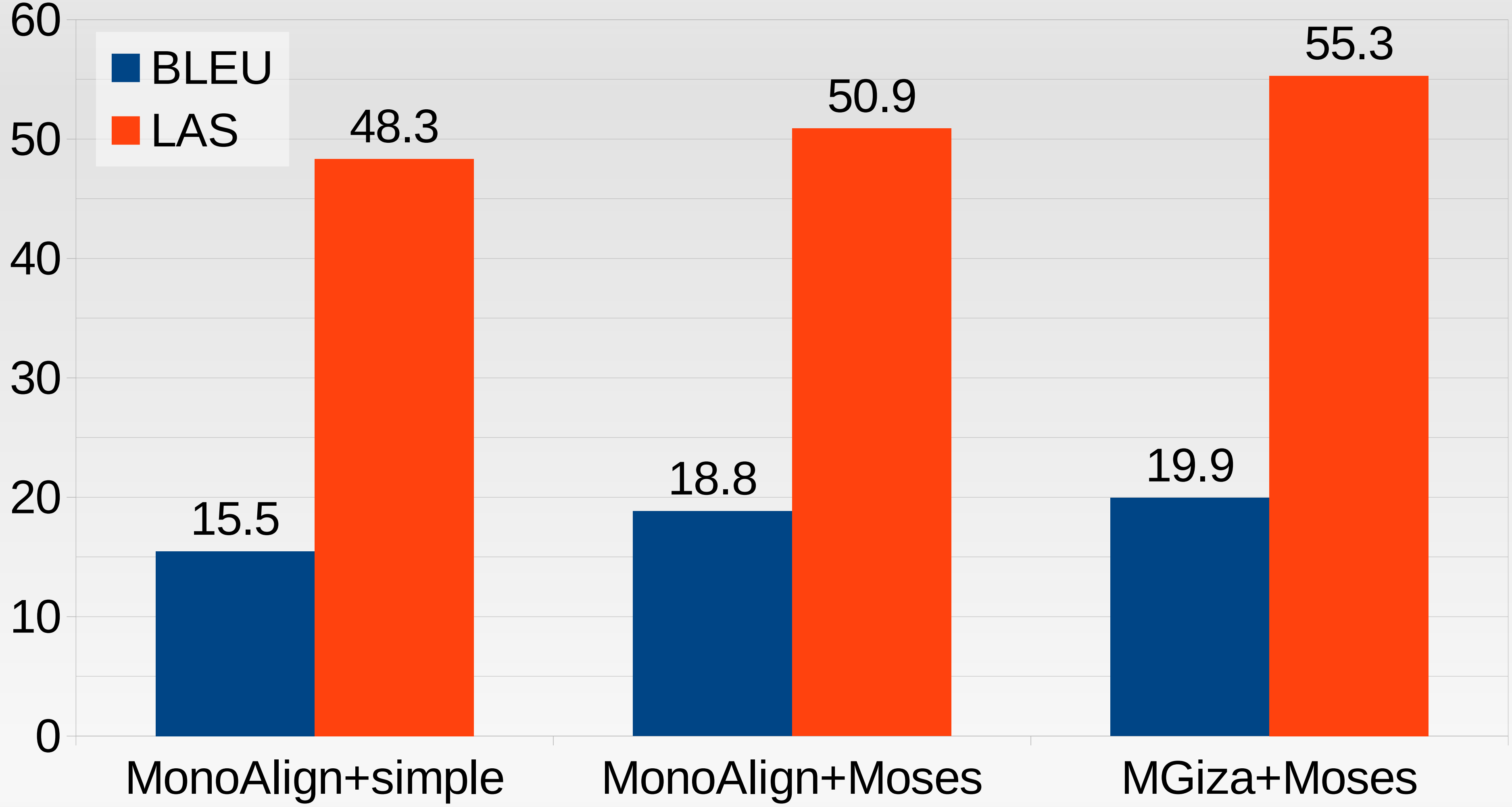
Tried various MT setups

- word-alignment and decoding systems
  - Giza++/MGiza++ with Moses, word-based setting
    - not SotA anymore but still very good and reliable
  - MonolingualGreedy Aligner (MP) / MonoAlign (DM) with simple single-best decoding
    - Jaro-Winkler, POS, position
  - MonoTrans (RR)
    - translation/guessing without parallel data
- also tried other combinations

Tried various MT setups

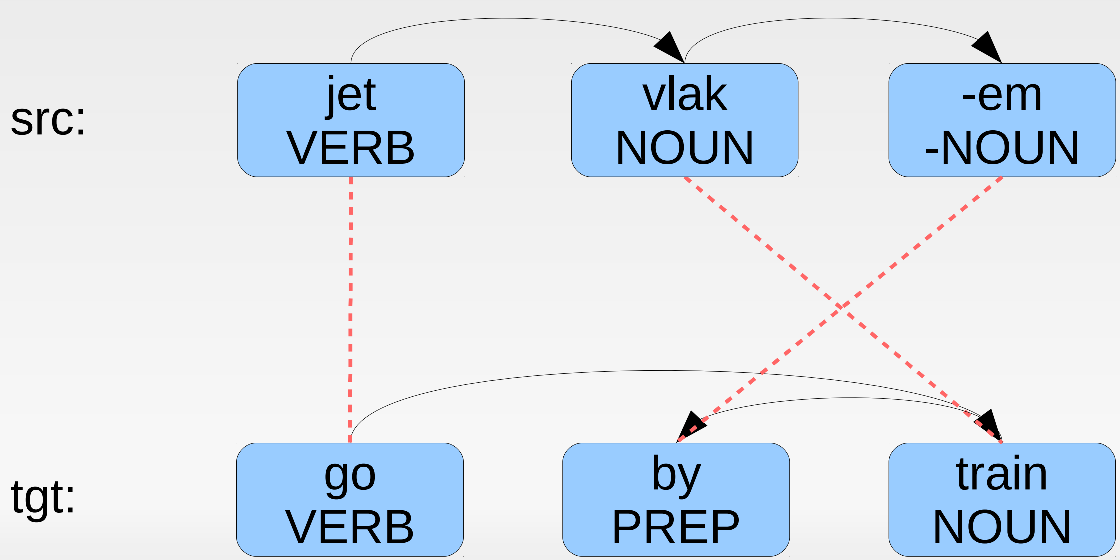
- word-alignment and decoding systems
  - **Giza++/MGiza++ with Moses, word-based setting**
    - not SotA anymore but still very good and reliable
  - MonolingualGreedy Aligner (MP) / MonoAlign (DM) with simple single-best decoding
    - Jaro-Winkler, POS, position
  - MonoTrans (RR)
    - translation/guessing without parallel data
- also tried other combinations

Various MT setups (12 lang pairs)



Tried various morphs/subwords

- morphs could get closer to 1:1 correspondence
  - joint segmentation and alignment? (Synder+, 2008)
- translation via morphs could do with less data
  - split rare complex words into frequent simple morphs



- complex issue
  - how to split?
  - how to parse?
  - how to label?
  - adds noise

Subwords in parsing

- splitting into subwords adds noise
  - similar words can get split differently
  - additional noise: affix/root classification
- still hard to achieve the 1:1 alignment
  - parallel data not sufficiently parallel
  - does not solve all phenomena
- root instead of original word, affixes as leaves
  - adds noise, does not bring improvements
  - automatic parse tree may be “invalid”

Bilingual word embeddings

- no improvement found under various setups
  - word2vec, fastText, SID-SGNS (Levy+, 2016)
- parser seems to rely on word identity a lot
  - embeddings useful only in tiny local neighbourhood
  - cannot exploit the full continuous vector space
  - fails if embeddings are transferred into “void”
    - summing/averaging/interpolating all bad
  - mediocre if same vectors used on both sides
    - why should be better than 1:1 MT?
    - MT has disambiguation, embeddings don't

Outline

Cross-lingual Transfer of Dependency Parsers

- Brief overview of the problem and a solution
- Why and how we parse text
- Without Machine Translation: Delex parsing
- How to do Machine Translation
- **How to choose the source language**
- How to combine multiple sources

Choosing the source language

- base: always use English as the source
  - not very wise (e.g. 30% instead of 60%)
- for given target, use source that:
  - is very similar
    - family, word order, auxiliaries, morphology...
    - multidimensional, interesting problem
  - has large-enough data
    - treebank, parallel data
    - not much research

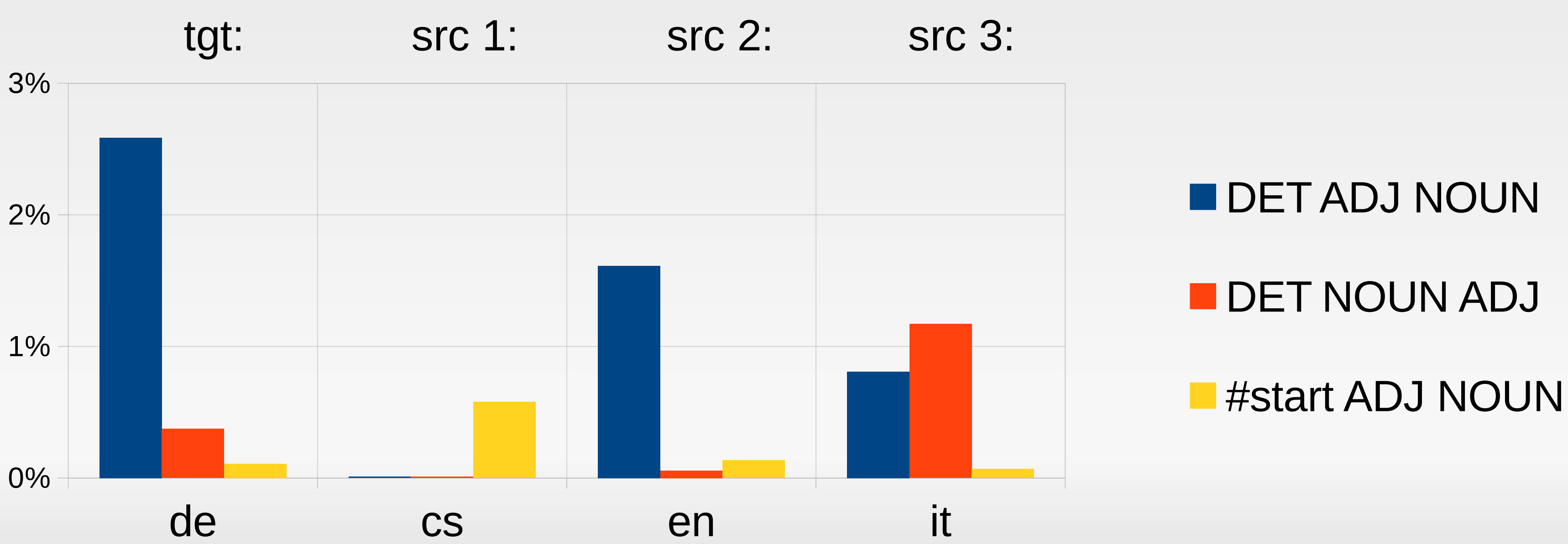
Source-target similarity

- typological properties from WALS (Naseem+, 2012)
  - language family, word order, morphology...
- distribution of POS tag ngrams (Rosa+, 2015)
  - similarity of word order and auxiliary usage
- lang-id based on character ngrams (Agić, 2017)
  - identify target language as one of the source langs.
- …combination of all of these (Agić, 2017)
  - possibly done separately for each sentence
- sentence weighting POS ngram LM (Søgaard+, 2012)

KL_{cpos^3} language similarity

- Kullback-Leibler divergence of POS trigram distributions

$$KL_{cpos^3}(tgt, src) = \sum_{\forall cpos^3 \in tgt} f_{tgt}(cpos^3) \cdot \log\left(\frac{f_{tgt}(cpos^3)}{f_{src}(cpos^3)}\right)$$



KL_{cpos^3} language similarity

- reasonable performance
  - identifies best source treebank in ~50% cases
  - less reliable on more distant language pairs
- requires POS-tagged target data
  - so far: only evaluated with gold POS and delex
  - future work: evaluate with cross-lingual POS
    - but results of (Agić, 2017) are very promising

Using the source-target similarity

- select best source
- weighted combination of multiple sources

Outline

<u>Cross-lingual Transfer of Dependency Parsers</u>

- Brief overview of the problem and a solution
- Why and how we parse text
- Without Machine Translation: Delex parsing
- How to do Machine Translation
- How to choose the source language
- **How to combine multiple sources**

Multilingual parser combination

- treebank concatenation (McDonald+, 2011)
- parse tree combination (Rosa+, 2015)
- parser model interpolation (Rosa+, 2015)
- …
- ±weighting by language similarity
- pre-existing: monolingual parser combination
  - Zeman+ (2005), Holan+ (2006), Sagae+ (2006), Green+ (2012), Green (2013)...
- note: older experiments (delex, unlabelled)

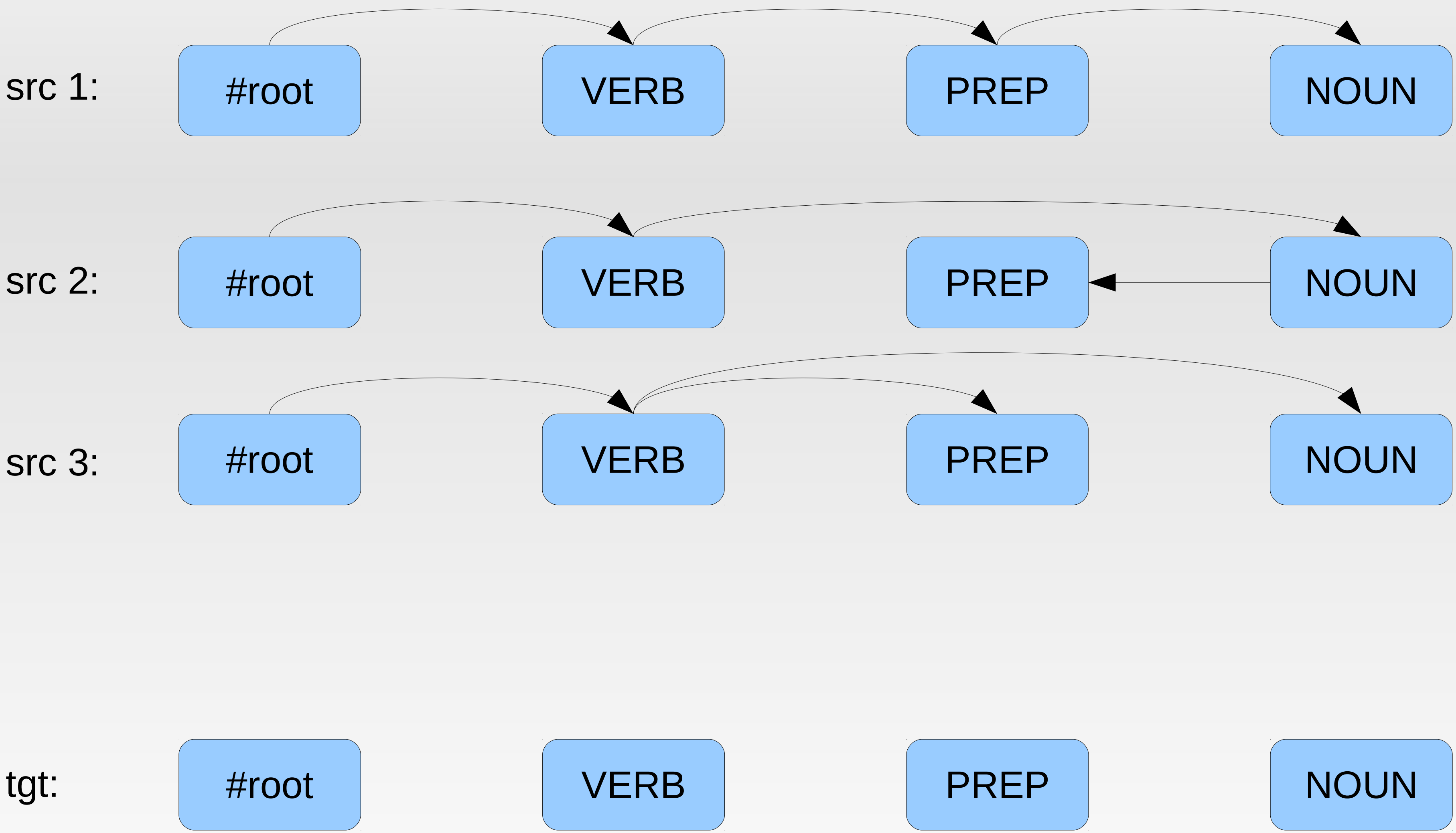
Treebank concatenation

- concatenate all source treebanks
  - delexicalized or after translation into target language
- train one parser on the multi-treebank
- apply the parser to the target text
- baseline method
  - weighting difficult (must modify training algorithm)
  - takes ages to train (huge data)
  - treebank influence proportional to its size
  - outcome = one standard parser (universal if delex)

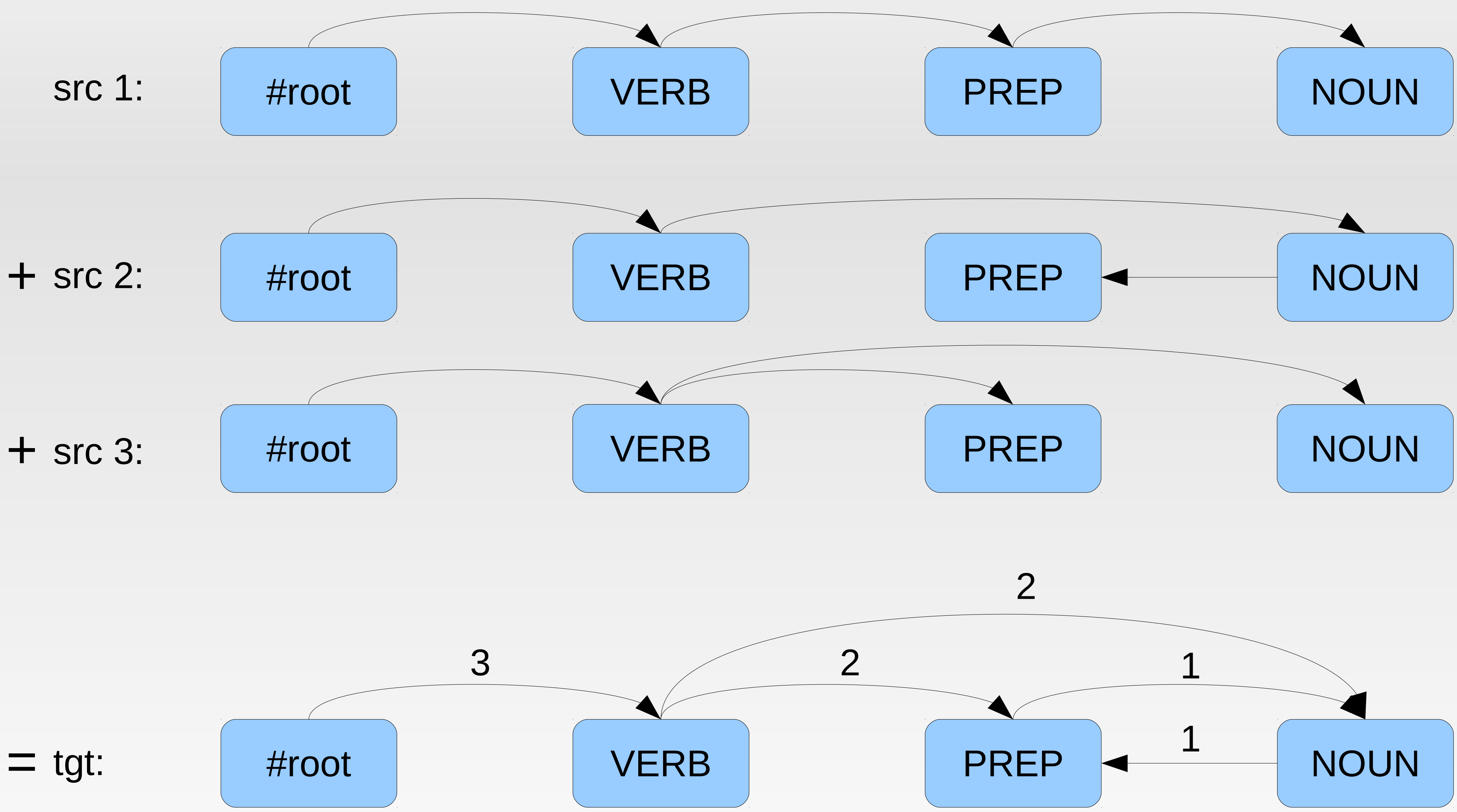
Parse tree combination

- train a separate parser for each source treebank
  - delexicalized or after translation into target language
- separately apply each parser to target text
- voting on edges & MST algorithm → final tree

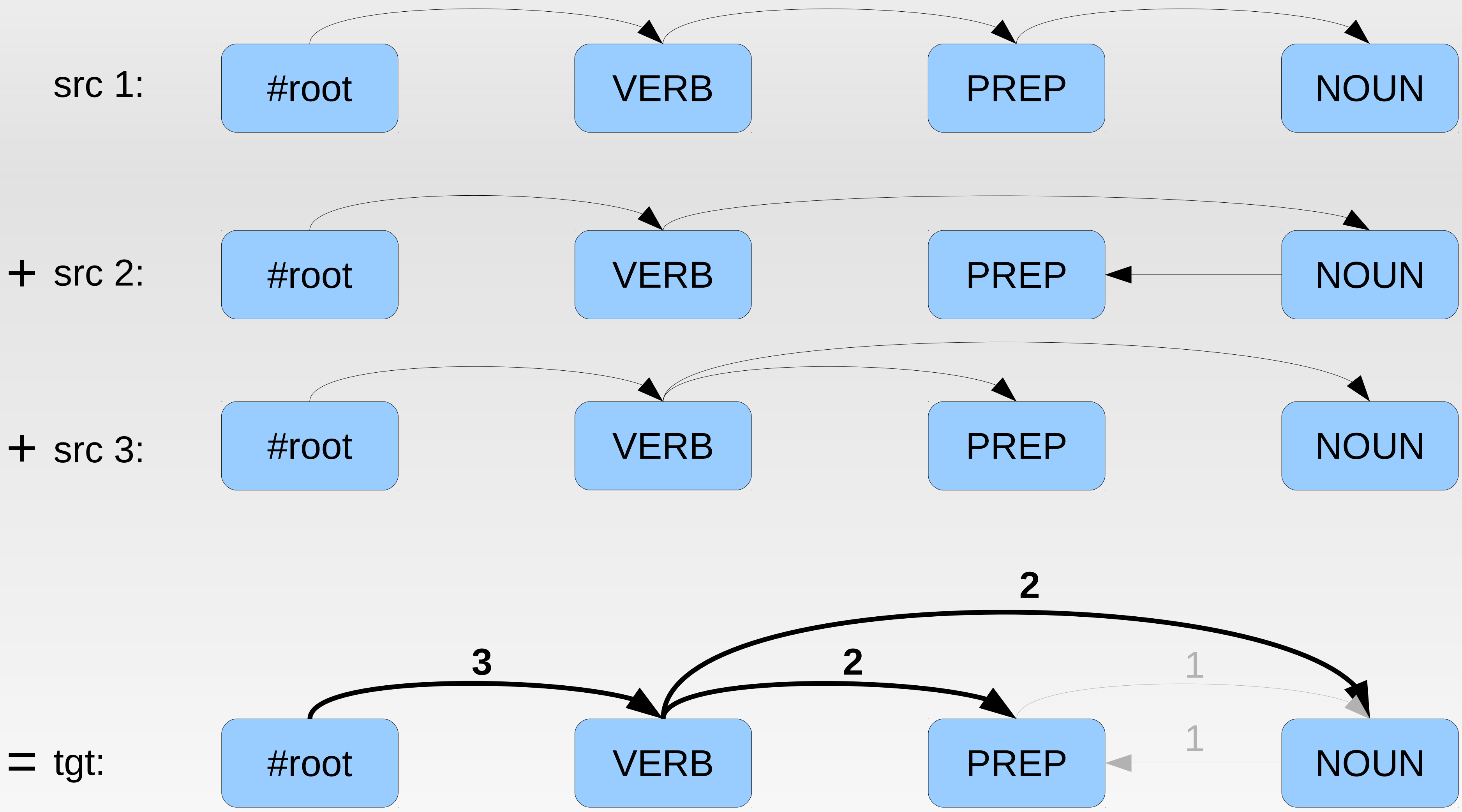
Parse tree combination



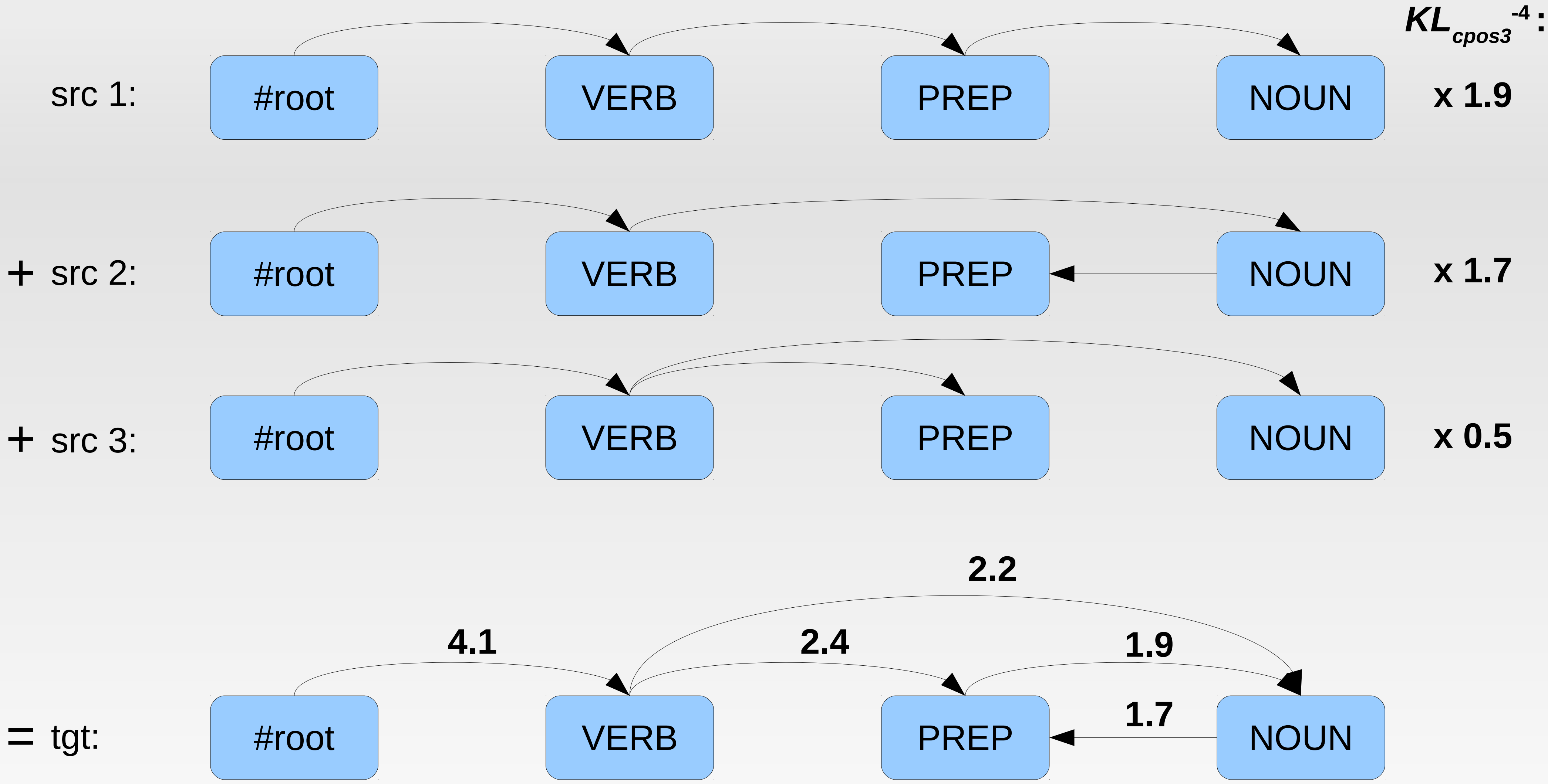
Parse tree combination



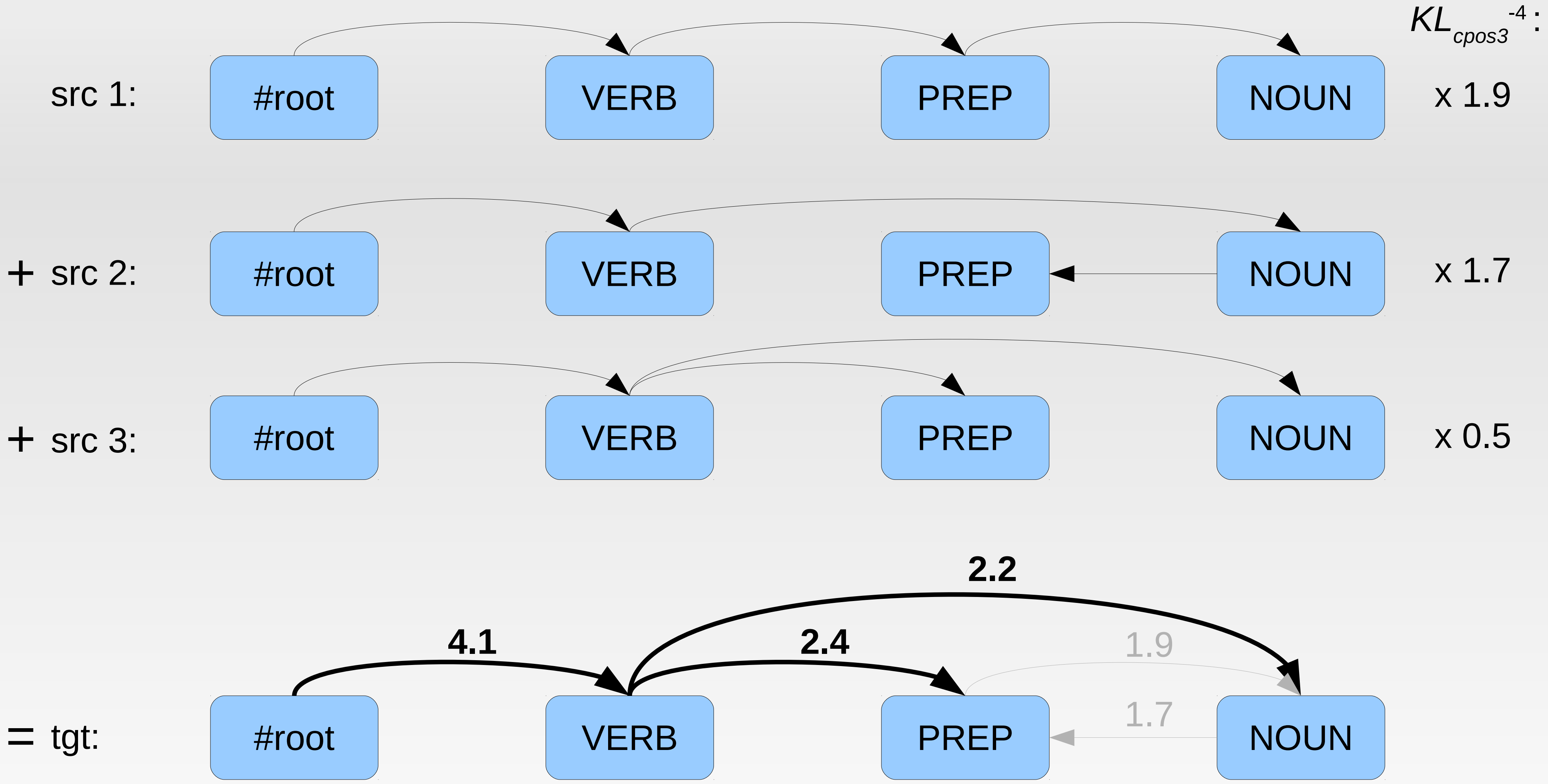
Parse tree combination



Weighted parse tree combination



Weighted parse tree combination



Parse tree combination

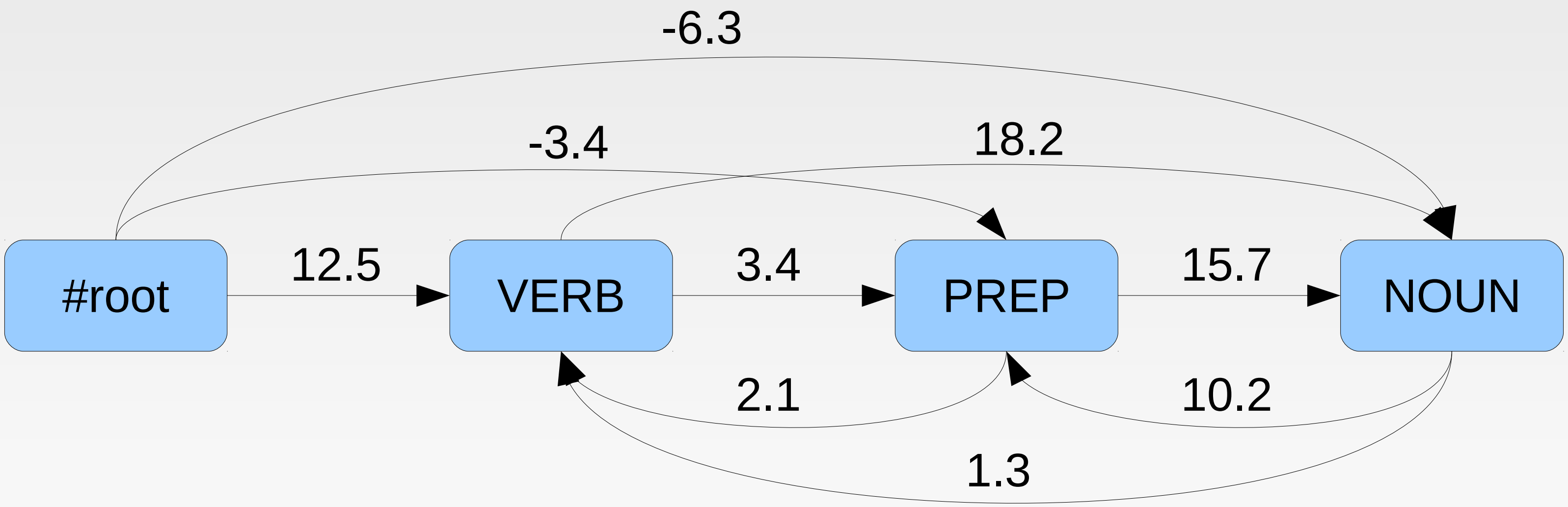
- train a separate parser for each source treebank
  - delexicalized or after translation into target language
- separately apply each parser to target text
- voting on edges & MST algorithm → final tree
- well-performing method
  - weighting easy
  - training naturally parallelizable
  - treebank size not leaking
  - outcome = N parsers

Parser model interpolation

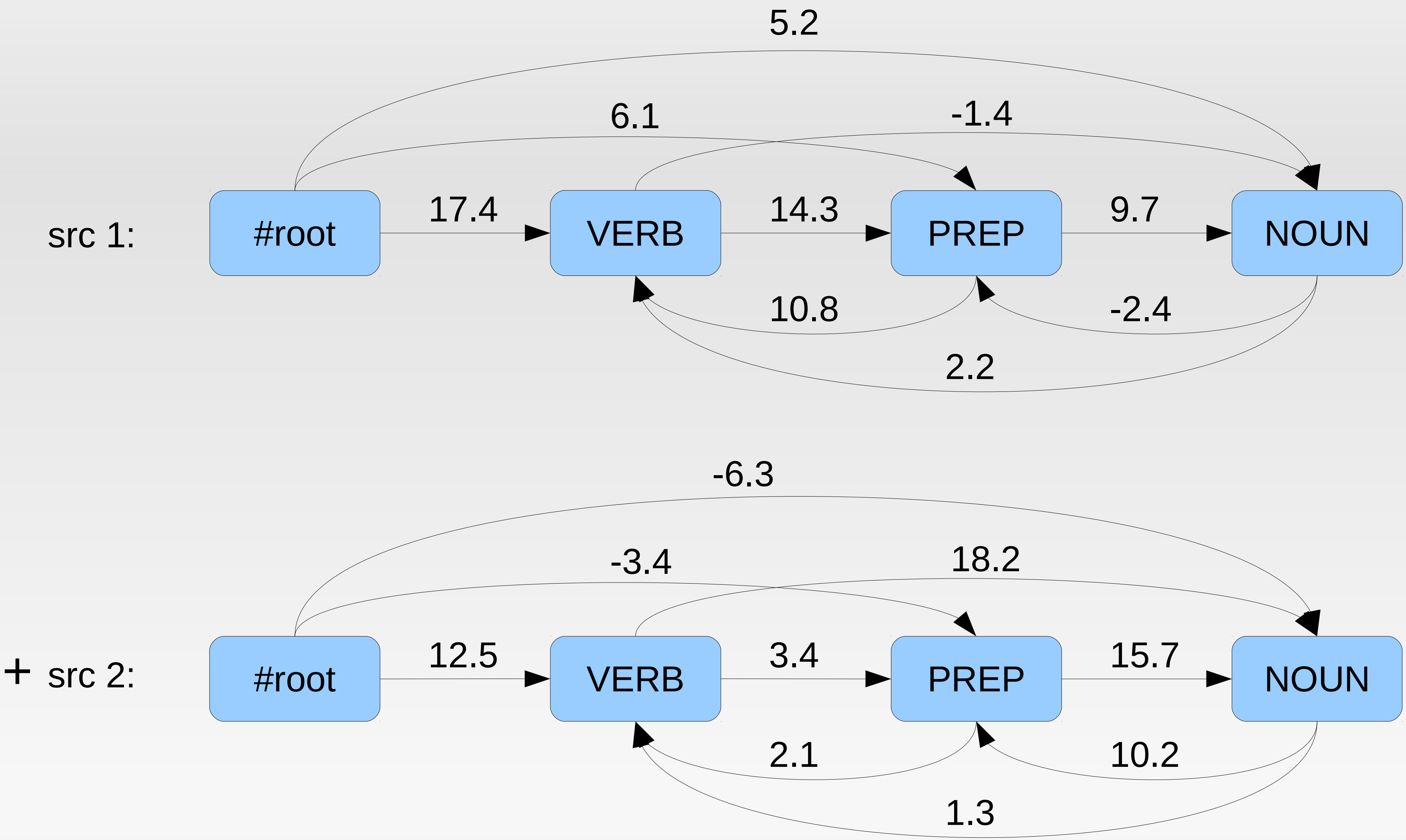
- train a separate parser for each source treebank
  - delexicalized or after translation into target language
- interpolate trained models into a combined model
- apply parser with combined model to target text

Parser model interpolation

- motivation: maybe the parser is more sure with some edges than other?
- the score assigned to the edge might show that
  - MSTParser before running the MST algorithm:



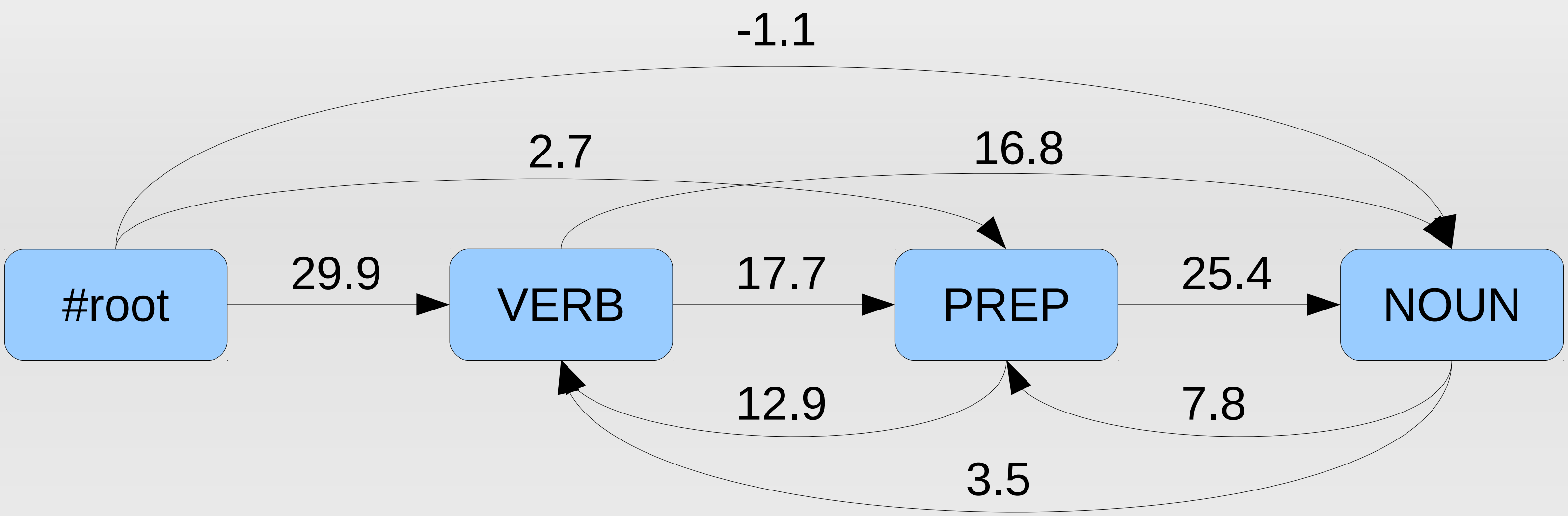
Parser model interpolation



- score normalization by standard deviation

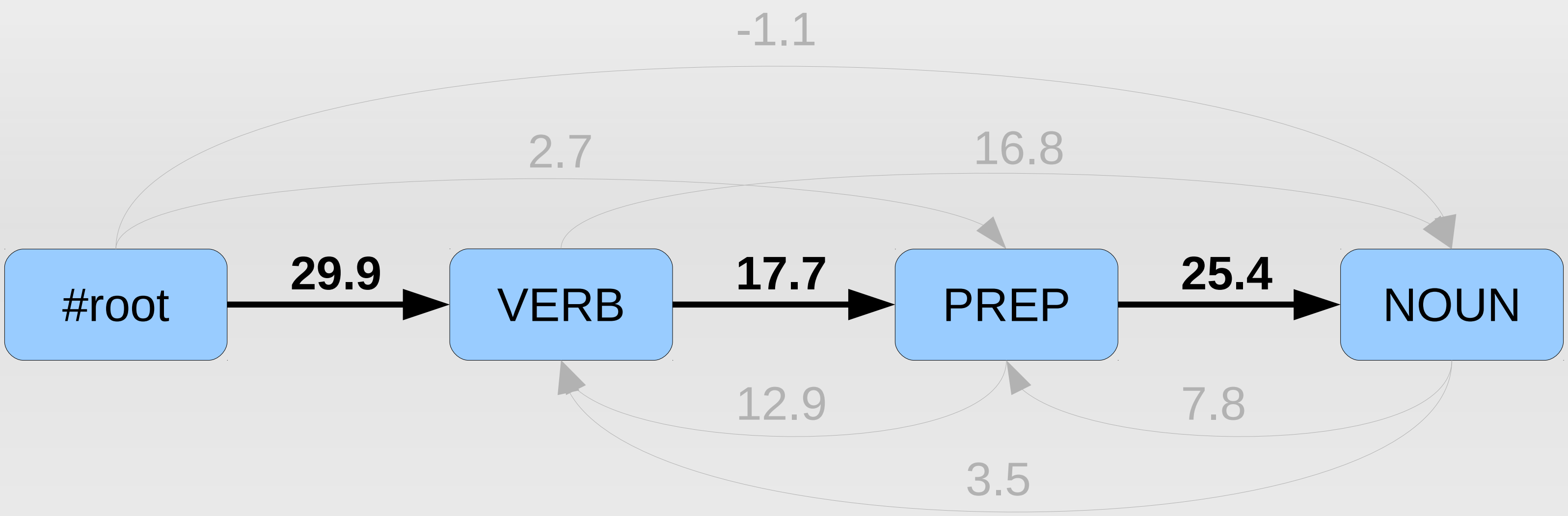
Parser model interpolation

= tgt (Σ):



Parser model interpolation

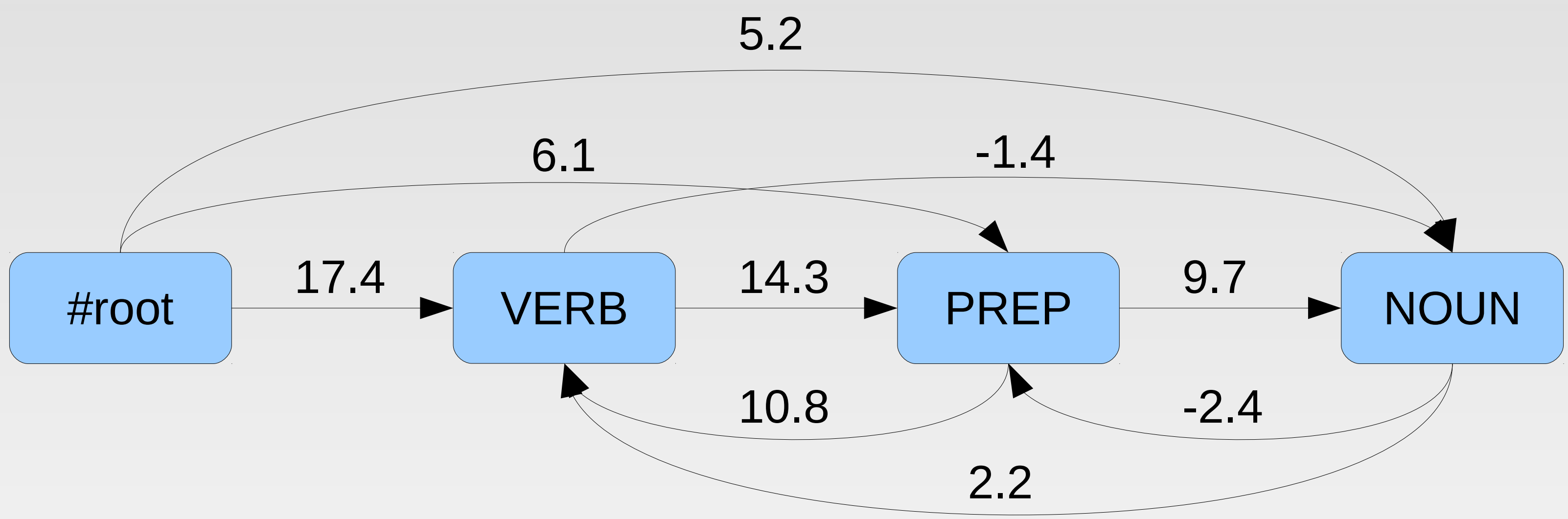
= tgt:



Weighted parser model interpol.

- multiply each edge score with KL_{cpos3}^{-4} (*tgt*,*src*)

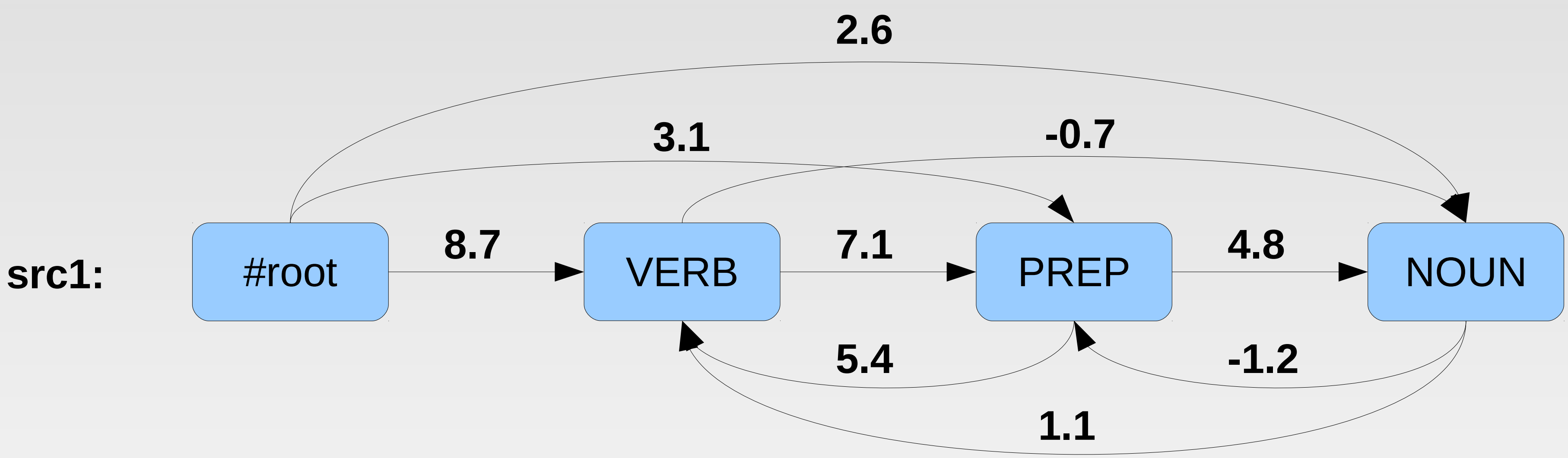
src1:



KL_{cpos3}^{-4} (*tgt*, *src1*) = 0.5

Weighted parser model interpol.

- multiply each edge score with $KL_{cpos3}^{-4}(tgt,src)$

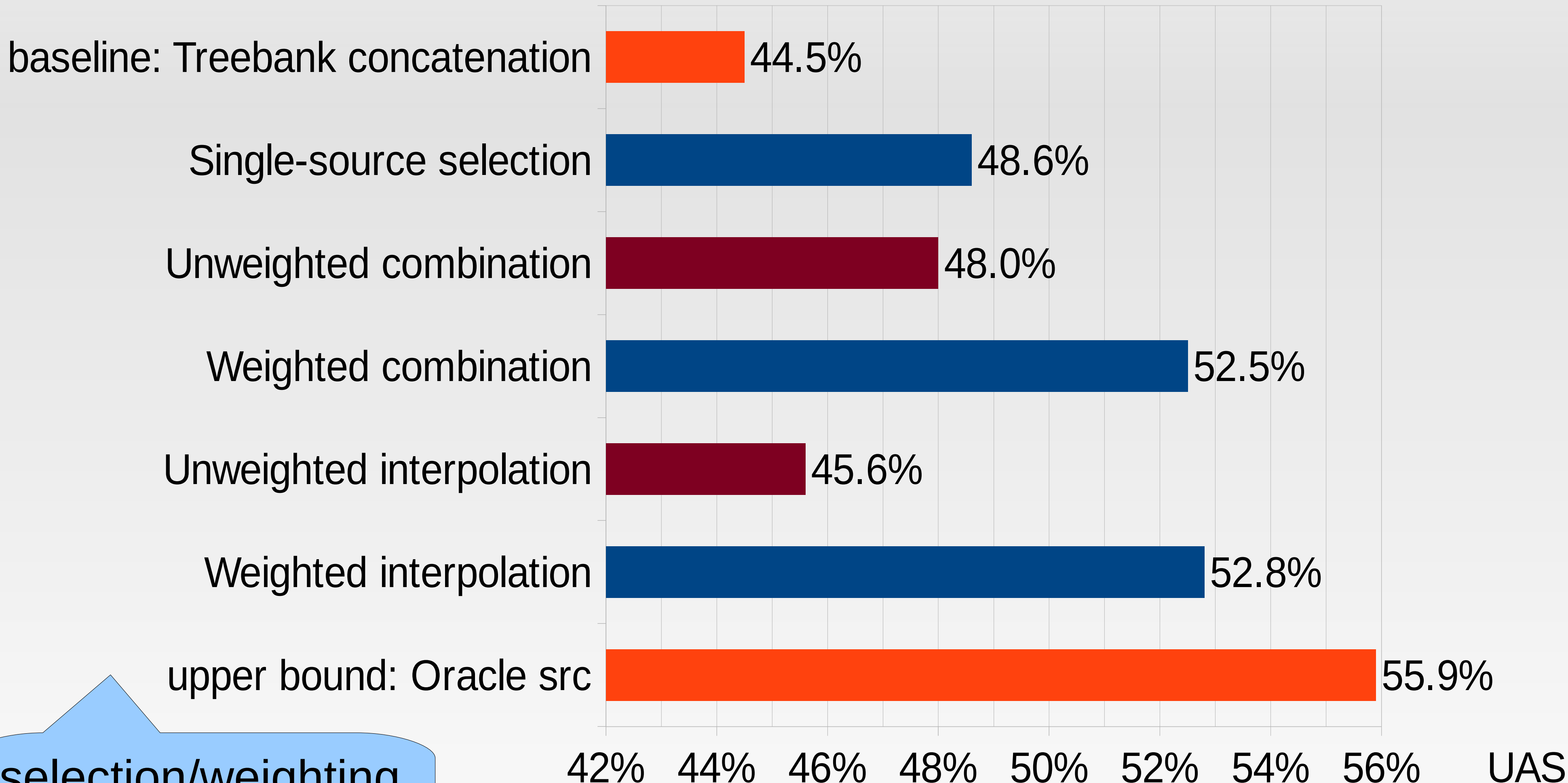


$KL_{cpos3}^{-4}(tgt, src1) = 0.5$

Parser model interpolation

- motivation: maybe the parser is more sure with some edges than other?
- the score assigned to the edge **might** show that
  - edge score ≠ parser confidence!
    - just a very rough estimate
  - better methods exist (Mejer+, 2012)
    - tree score drop when <u>the edge</u> forbidden
    - % of trees with <u>the edge</u> in k-best, weighted
    - % of trees with <u>the edge</u> in K sampled models
    - …more accurate, but slower and less practical...

Average UAS over 18 test TBs



selection/weighting using KL_{cpos}^3

Conclusion

- Parsing of low-resourced natural languages
- Delexicalized parsing → unrealistic
- Lexicalization via MT → not straightforward
- Multiple sources available → select or combine
- Future work:
  - higher-quality MT (reordering, N:N, 1:N, M:N)
  - lexicalized source selection/weighting (no gold POS)
  - combine best setups together
  - finish thesis :-)

Thank you for your attention

Rudolf Rosa
rosa@ufal.mff.cuni.cz

Cross-lingual Transfer of Dependency Parsers

Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

http://ufal.mff.cuni.cz/rudolf-rosa/