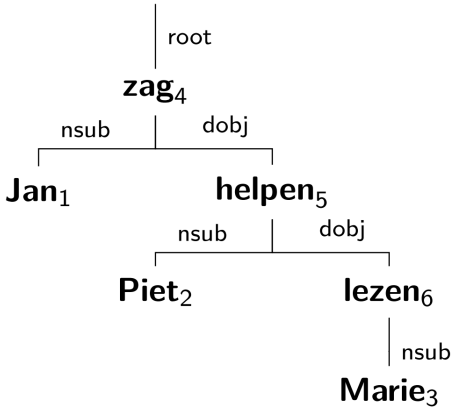
Parsing of lexicalised LCFRS via supertagging

Alex Ivliev

TU Dresden

Goal

$_0$ Jan $_1$ Piet $_2$ Marie $_3$ zag $_4$ helpen $_5$ lezen $_6$



Structure

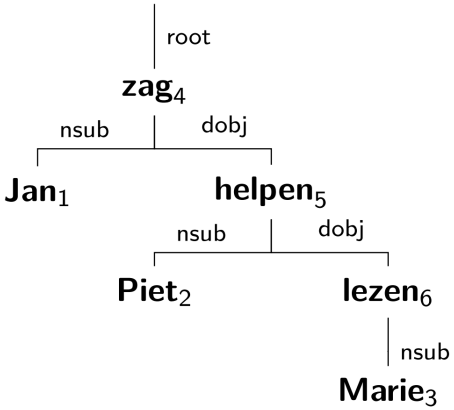
1. Grammar extraction and parsing
2. Supertagging
3. BERT
4. BERT for supertagging
5. Experimental results

Approach

- Corpus: Hamburg Dependency Treebank
- sourced from the German news site heise.de
- 68,801 training sentences, 17,028 test sentences

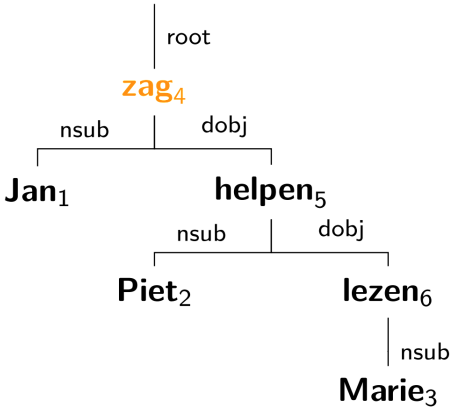
Grammar extraction - Kuhlmann und Satta, 2009

Jan Piet Marie zag helpen lezen



Grammar extraction - Kuhlmann und Satta, 2009

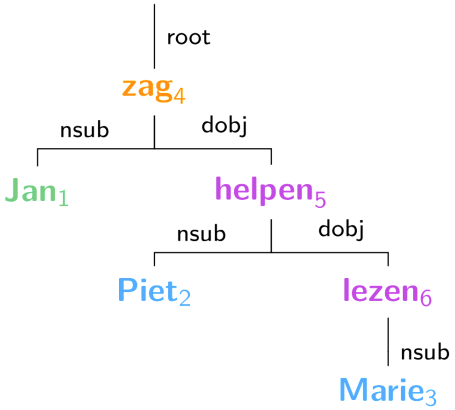
Jan Piet Marie zag helpen lezen



root( ) → nsub( )dobj( )

Grammar extraction - Kuhlmann und Satta, 2009

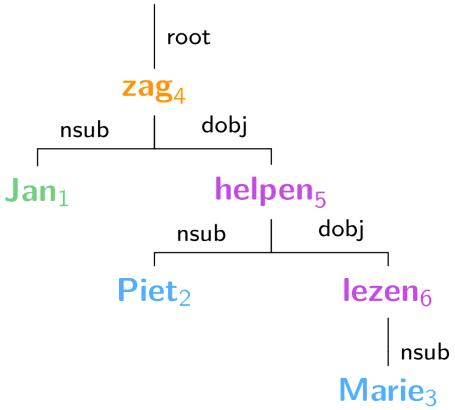
Jan Piet Marie zag helpen lezen



root( ) $\rightarrow$ nsub(x_1)dobj(x_2, x_3)

Grammar extraction - Kuhlmann und Satta, 2009

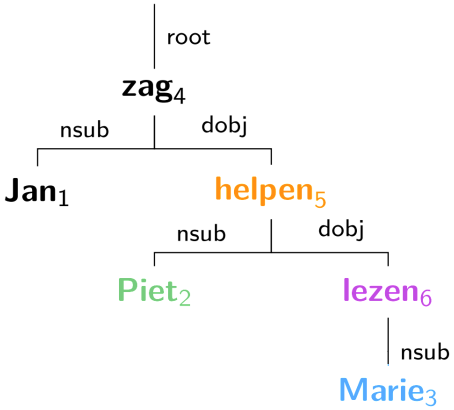
Jan Piet Marie zag helpen lezen



$$\text{root}(x_1 x_2 \text{zag} x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$$

Grammar extraction - Kuhlmann und Satta, 2009

Jan Piet Marie zag helpen lezen

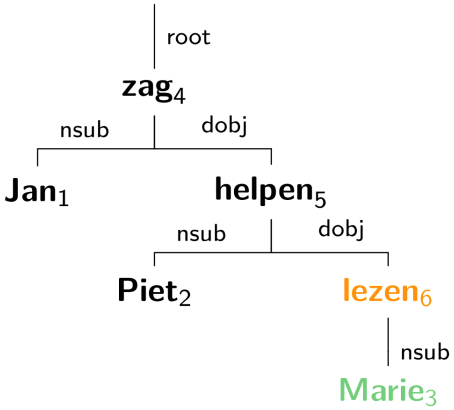


$$\text{root}(x_1 x_2 \text{zag} x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$$

$$\text{dobj}(x_1 x_2, \text{helpen} x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$$

Grammar extraction - Kuhlmann und Satta, 2009

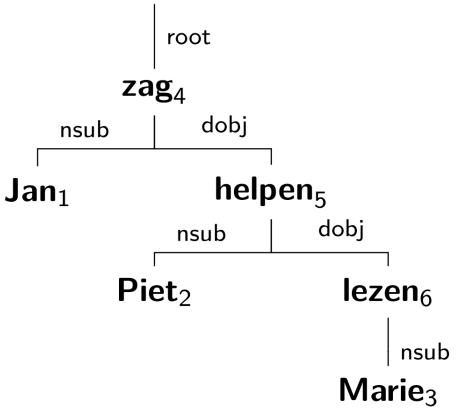
Jan Piet Marie zag helpen lezen



$$\text{root}(x_1 x_2 \text{zag} x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$$
$$\text{dobj}(x_1 x_2, \text{helpen} x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$$
$$\text{dobj}(x_1, \text{lezen}) \rightarrow \text{nsub}(x_1)$$

Grammar extraction - Kuhlmann und Satta, 2009

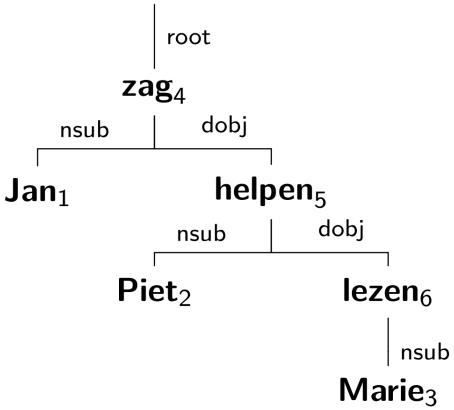
Jan Piet Marie zag helpen lezen



$$\text{root}(x_1 x_2 \text{zag} x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$$
$$\text{dobj}(x_1 x_2, \text{helpen} x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$$
$$\text{dobj}(x_1, \text{lezen}) \rightarrow \text{nsub}(x_1)$$
$$\text{nsub}(\text{Jan}) \rightarrow \varepsilon$$
$$\text{nsub}(\text{Piet}) \rightarrow \varepsilon$$
$$\text{nsub}(\text{Marie}) \rightarrow \varepsilon$$

Grammar extraction - Kuhlmann und Satta, 2009

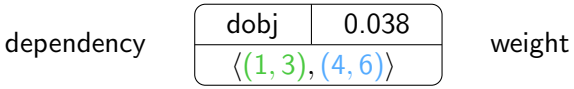
Jan Piet Marie zag helpen lezen



$$1.0\ \mathrm{root}(x_1 x_2 \mathrm{zag} x_3) \rightarrow \mathrm{nsub}(x_1)\mathrm{dobj}(x_2, x_3)$$
$$0.7\ \mathrm{dobj}(x_1 x_2, \mathrm{helpen} x_3) \rightarrow \mathrm{nsub}(x_1)\mathrm{dobj}(x_2, x_3)$$
$$0.3\ \mathrm{dobj}(x_1, \mathrm{lezen}) \rightarrow \mathrm{nsub}(x_1)$$
$$0.1\ \mathrm{nsub}(\mathrm{Jan}) \rightarrow \varepsilon$$
$$0.3\ \mathrm{nsub}(\mathrm{Piet}) \rightarrow \varepsilon$$
$$0.6\ \mathrm{nsub}(\mathrm{Marie}) \rightarrow \varepsilon$$

Parsing - Nederhof, 2003

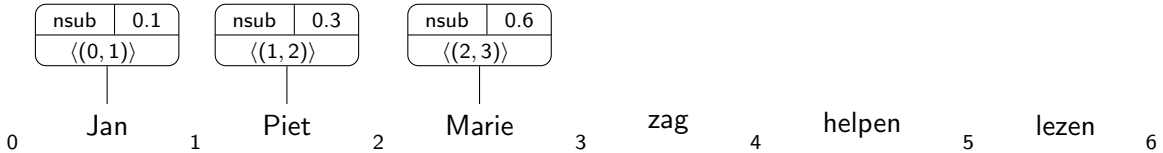
Parseitem:



0 Jan 1 Piet 2 Marie 3 zag 4 helpen 5 lezen 6

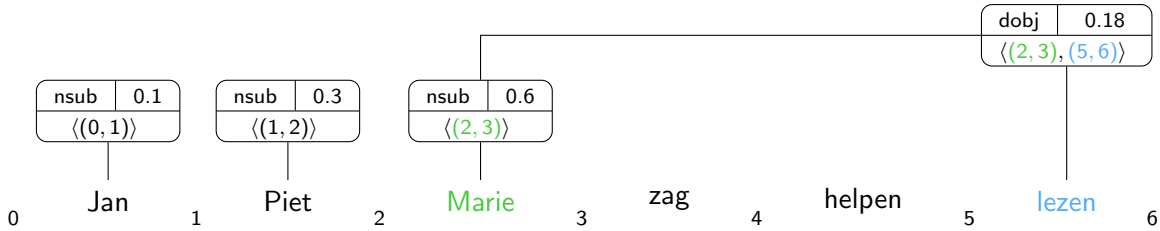
Parsing - Nederhof, 2003

0.1: nsub(Jan) $\rightarrow \varepsilon$ 0.3: nsub(Piet) $\rightarrow \varepsilon$ 0.6: nsub(Marie) $\rightarrow \varepsilon$



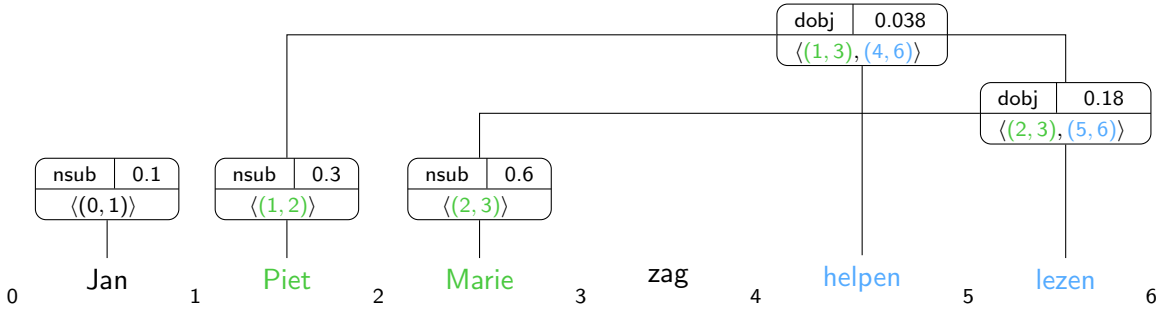
Parsing - Nederhof, 2003

0.3: $\text{dobj}(x_1, \text{lezen}) \rightarrow \text{nsub}(x_1)$



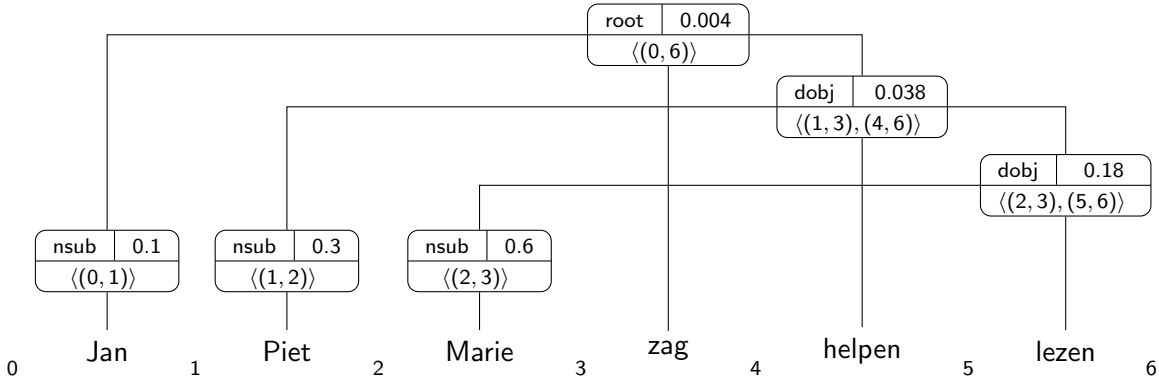
Parsing - Nederhof, 2003

0.7: $\text{dobj}(x_1 x_2, \text{helpen} x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$

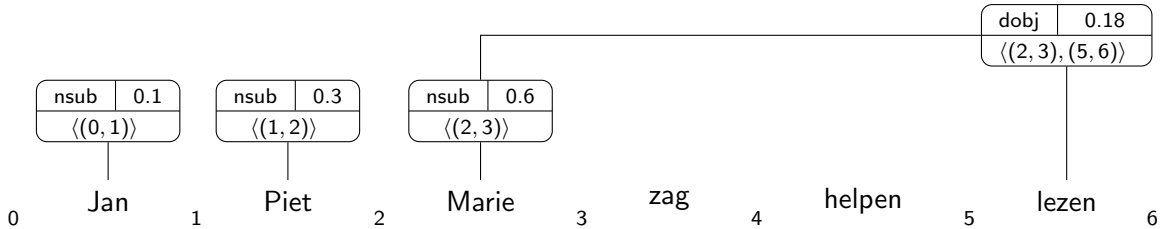


Parsing - Nederhof, 2003

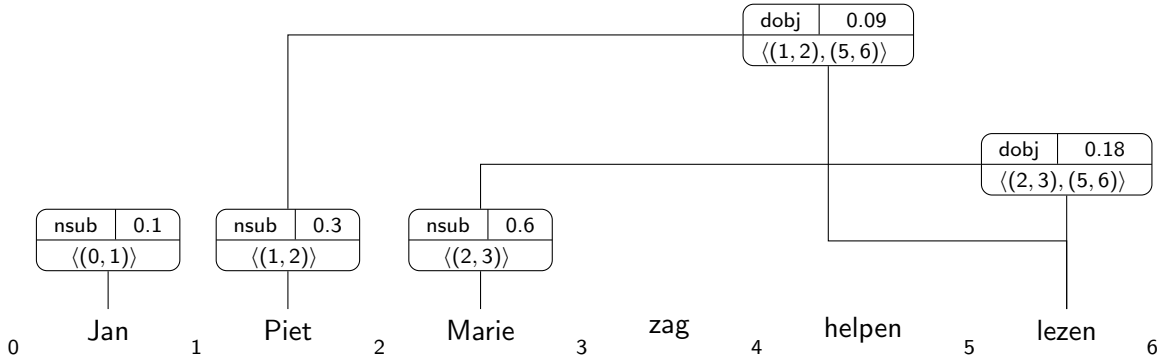
1.0: $\text{root}(x_1 x_2 \text{zag} x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$



$$0.3\text{: } \mathrm{dobj}(x_1, \mathrm{lezen}) \rightarrow \mathrm{nsub}(x_1)$$



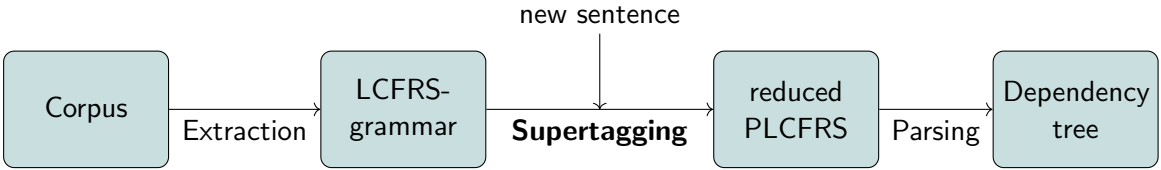
$0.3\colon\ \mathrm{dobj}(x_1, \mathrm{lezen}) \rightarrow \mathrm{nsub}(x_1)$



Supertagging - Joshi, Srinivas, 1994

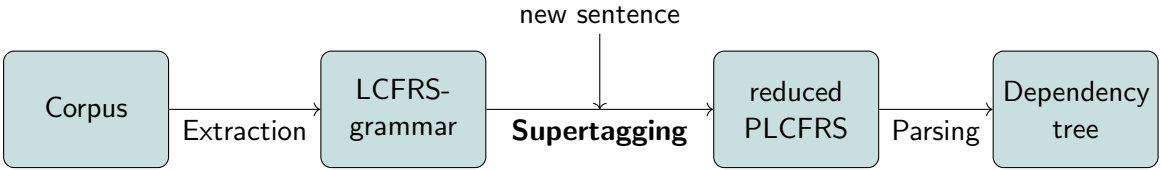
- Problem: Parsing is expensive for huge grammars
- Solution: Pre-selection of small number of promising rules

Supertagging - Joshi, Srinivas, 1994



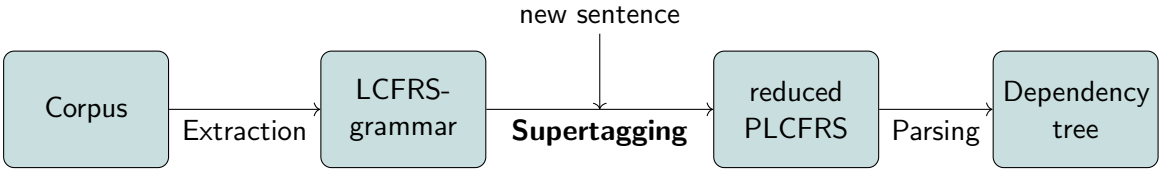
- Problem: Parsing is expensive for huge grammars
- Solution: Pre-selection of small number of promising rules

Supertagging - Joshi, Srinivas, 1994



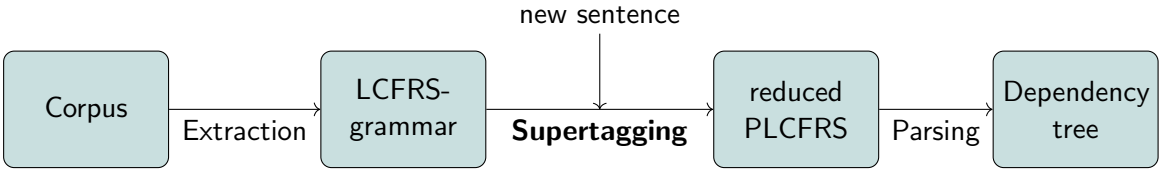
Jan	$\text{nsub}(\text{Jan}) \to \varepsilon$	$\text{subj}(\text{Jan}) \to \varepsilon$
Piet	$\text{dobj}(\text{Piet}) \to \varepsilon$	$\text{nsub}(\text{Piet}) \to \varepsilon$
Marie	$\text{subj}(x_1, \text{Marie}) \to \text{nsub}(x_1)$	$\text{nsub}(\text{Marie}) \to \varepsilon$
zag	$\text{root}(x_1 x_2 \text{zag} x_3) \to \text{nsub}(x_1)\text{dobj}(x_2, x_3)$	$\text{aux}(x_1 \text{zag} x_2) \to \text{dobj}(x_1, x_2)$
helpen	$\text{dobj}(x_1 x_2, \text{helpen} x_3) \to \text{nsub}(x_1)\text{dobj}(x_2, x_3)$	$\text{dobj}(x_1, \text{helpen}) \to \text{nsub}(x_1)$
lezen	$\text{dobj}(\text{lezen}, x_1) \to \text{nsub}(x_1)$	$\text{dobj}(x_1, \text{lezen}) \to \text{nsub}(x_1)$

Supertagging - Joshi, Srinivas, 1994



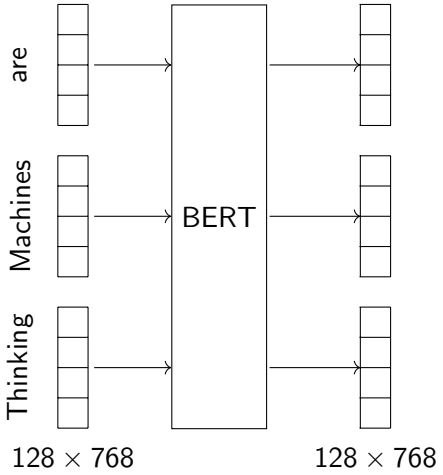
Jan	$\text{nsub}(\text{David}) \to \varepsilon$	$\text{subj}(\text{Jan}) \to \varepsilon$
Piet	$\text{dobj}(\text{Piet}) \to \varepsilon$	$\text{nsub}(\text{Piet}) \to \varepsilon$
Marie	$\text{subj}(x_1, \text{Marie}) \to \text{nsub}(x_1)$	$\text{nsub}(\text{Marie}) \to \varepsilon$
zag	$\text{root}(x_1 x_2 \text{zag} x_3) \to \text{nsub}(x_1)\text{dobj}(x_2, x_3)$	$\text{aux}(x_1 \text{zag} x_2) \to \text{dobj}(x_1, x_2)$
helpen	$\text{dobj}(x_1 x_2, \text{helpen} x_3) \to \text{nsub}(x_1)\text{dobj}(x_2, x_3)$	$\text{dobj}(x_1, \text{helpen}) \to \text{nsub}(x_1)$
lezen	$\text{dobj}(\text{lezen}, x_1) \to \text{nsub}(x_1)$	$\text{dobj}(x_1, \text{lezen}) \to \text{nsub}(x_1)$

Supertagging - Joshi, Srinivas, 1994



Jan	$\text{nsub}(\langle \text{T} \rangle) \to \varepsilon$	$\text{subj}(\langle \text{T} \rangle) \to \varepsilon$
Piet	$\text{dobj}(\langle \text{T} \rangle) \to \varepsilon$	$\text{nsub}(\langle \text{T} \rangle) \to \varepsilon$
Marie	$\text{subj}(x_1, \langle \text{T} \rangle) \to \text{nsub}(x_1)$	$\text{nsub}(\langle \text{T} \rangle) \to \varepsilon$
zag	$\text{root}(x_1 x_2 \langle \text{T} \rangle x_3) \to \text{nsub}(x_1)\text{dobj}(x_2, x_3)$	$\text{aux}(x_1 \langle \text{T} \rangle x_2) \to \text{dobj}(x_1, x_2)$
helpen	$\text{dobj}(x_1 x_2, \langle \text{T} \rangle x_3) \to \text{nsub}(x_1)\text{dobj}(x_2, x_3)$	$\text{dobj}(x_1, \langle \text{T} \rangle) \to \text{nsub}(x_1)$
lezen	$\text{dobj}(\langle \text{T} \rangle, x_1) \to \text{nsub}(x_1)$	$\text{dobj}(x_1, \langle \text{T} \rangle) \to \text{nsub}(x_1)$

BERT - Devlin et al., 2019



BERT Pre-Training

Network is trained on

1. Masked Language Model
   The man went to the [redacted] to buy a [redacted] of milk.

BERT Pre-Training

Network is trained on

1. Masked Language Model
   The man went to the store to buy a gallon of milk.

BERT Pre-Training

Network is trained on

1. Masked Language Model
   The man went to the store to buy a gallon of milk.
2. Next Sentence Prediction
   A: The man went to the store.
   B: He bought a gallon of milk.

   A: The man went to the store.
   B: Penguins are flightless.

BERT Pre-Training

Network is trained on

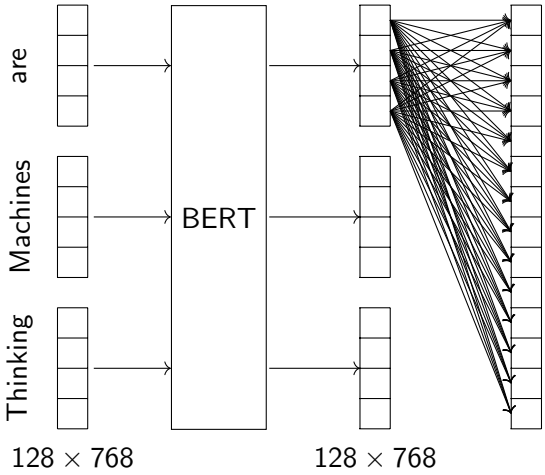
1. Masked Language Model
   The man went to the store to buy a gallon of milk.
2. Next Sentence Prediction
   A: The man went to the store.
   B: He bought a gallon of milk.

   A: The man went to the store.
   B: Penguins are flightless.

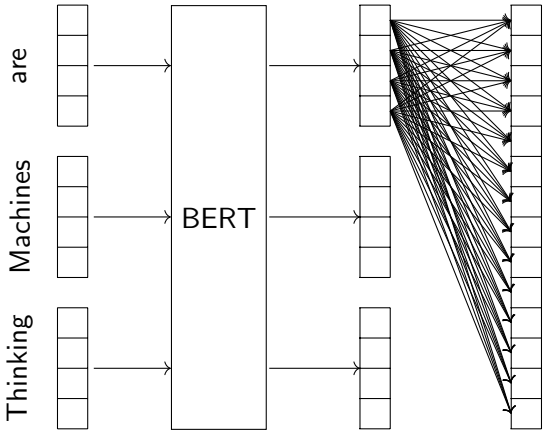
→ Pretrained model has an understanding of language

→ Pretrained model is available online

BERT - Finetuning

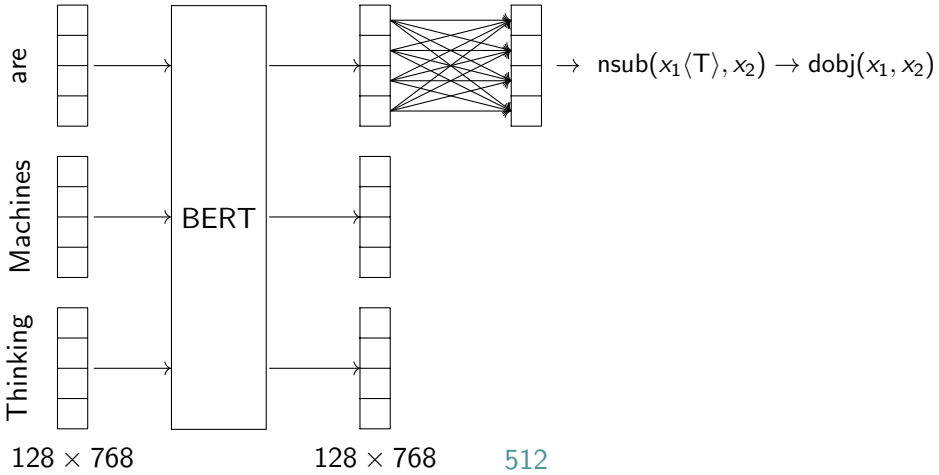


BERT - Finetuning

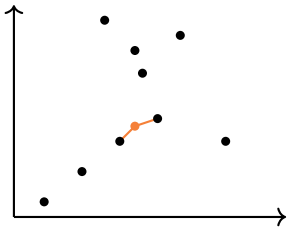


$\text{nsub}(x_1\langle \text{T}\rangle, x_2) \rightarrow \text{dobj}(x_1, x_2)$
$\text{root}(x_1 x_2\langle \text{T}\rangle x_3) \rightarrow \text{dobj}(x_1, x_2)\text{nsub}(x_3)$
$\text{nsub}(\langle \text{T}\rangle) \rightarrow \varepsilon$
$\ldots$

BERT - Finetuning



Embedding of LCFRS-rules



- Assign a vector of real numbers to each LCFRS-rule
- BERT outputs a vector
- Choose the n closest vectors

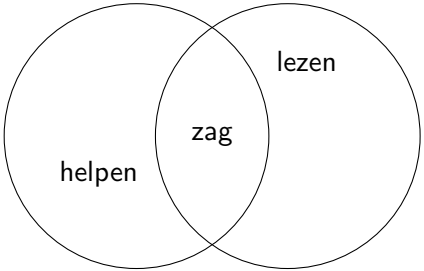
$\rightarrow$ Nearby vectors should represent "similar" rules

Similarity measures - Tokens

$r = \text{dobj}(x_1\langle T\rangle, x_2) \rightarrow \text{nsub}(x_1, x_2)$

$\rightarrow T(r) = \{\text{zag}, \text{helpen}, \text{lezen}\}$

- $\text{dobj}(x_1\text{zag}, x_2) \rightarrow \text{nsub}(x_1, x_2)$
- $\text{dobj}(x_1\text{helpen}, x_2) \rightarrow \text{nsub}(x_1, x_2)$
- $\text{dobj}(x_1\text{lezen}, x_2) \rightarrow \text{nsub}(x_1, x_2)$



Similarity measures - Tokens

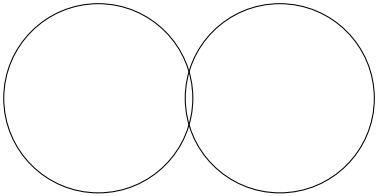
$r = \text{dobj}(x_1\langle T\rangle, x_2) \rightarrow \text{nsub}(x_1, x_2)$

$\rightarrow T(r) = \{\text{zag}, \text{helpen}, \text{lezen}\}$

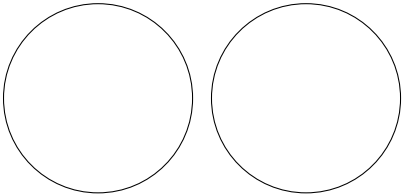
$\text{dobj}(x_1\text{zag}, x_2) \rightarrow \text{nsub}(x_1, x_2)$

$\text{dobj}(x_1\text{helpen}, x_2) \rightarrow \text{nsub}(x_1, x_2)$

$\text{dobj}(x_1\text{lezen}, x_2) \rightarrow \text{nsub}(x_1, x_2)$

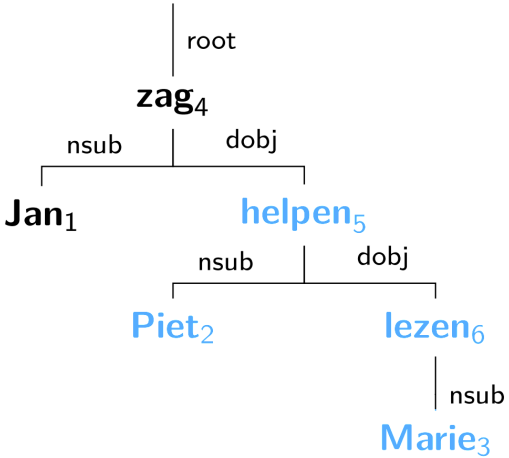


1%



99%

Similarity measures - Cover



$\rightarrow r = \text{dobj}(x_1 x_2, \langle \text{T} \rangle x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$

$C(r) \supseteq \{\text{helpen}, \text{Piet}, \text{lezen}, \text{Marie}\}$

Similarity measures - String distances

$\mathsf{dobj}(x_1, \langle \mathsf{T} \rangle x_2 x_3) \rightarrow \mathsf{nsub}(x_1)\mathsf{dobj}(x_2, x_3)$
$\mathsf{dobj}(x_1, x_2 \langle \mathsf{T} \rangle x_3) \rightarrow \mathsf{nsub}(x_1)\mathsf{dobj}(x_2, x_3)$

vs.

$\mathsf{root}(x_1 x_2 x_3, \langle \mathsf{T} \rangle) \rightarrow \mathsf{aux}(x_1, x_2)\mathsf{nsub}(x_3)$
$\mathsf{dobj}(\langle \mathsf{T} \rangle x_1, x_2) \rightarrow \mathsf{nsub}(x_1)\mathsf{det}(x_2)$

Similarity measures - String distances

$\mathsf{dobj}(x_1, \langle \mathsf{T} \rangle x_2 x_3) \rightarrow \mathsf{nsub}(x_1)\mathsf{dobj}(x_2, x_3)$
$\mathsf{dobj}(x_1, x_2 \langle \mathsf{T} \rangle x_3) \rightarrow \mathsf{nsub}(x_1)\mathsf{dobj}(x_2, x_3)$

vs.

$\mathsf{root}(x_1 x_2 x_3, \langle \mathsf{T} \rangle) \rightarrow \mathsf{aux}(x_1, x_2)\mathsf{nsub}(x_3)$
$\mathsf{dobj}(\langle \mathsf{T} \rangle x_1, x_2) \rightarrow \mathsf{nsub}(x_1)\mathsf{det}(x_2)$

String of rule: $\mathsf{A}(x_1 x_2 \langle \mathsf{T} \rangle, x_3) \rightarrow \mathsf{B}(x_1)\mathsf{C}(x_2, x_3) \rightsquigarrow \mathsf{BC}\langle \mathsf{T} \rangle, \mathsf{C}$

Similarity measures - String distances

$\text{dobj}(x_1, \langle \text{T} \rangle x_2 x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$
$\text{dobj}(x_1, x_2 \langle \text{T} \rangle x_3) \rightarrow \text{nsub}(x_1)\text{dobj}(x_2, x_3)$

vs.

$\text{root}(x_1 x_2 x_3, \langle \text{T} \rangle) \rightarrow \text{aux}(x_1, x_2)\text{nsub}(x_3)$
$\text{dobj}(\langle \text{T} \rangle x_1, x_2) \rightarrow \text{nsub}(x_1)\text{det}(x_2)$

String of rule: $\text{A}(x_1 x_2 \langle \text{T} \rangle, x_3) \rightarrow \text{B}(x_1)\text{C}(x_2, x_3) \rightsquigarrow \text{BC}\langle \text{T} \rangle,\text{C}$

Bigram	BC⟨T⟩,C	⟨T⟩,CBCB	Δ
BC	1	1	0
C⟨T⟩	1	0	1
⟨T⟩,	1	1	0
,C	1	1	0
CB	0	2	2

Similarity measures - String distances

$\mathrm{dobj}(x_1, \langle \mathrm{T} \rangle x_2 x_3) \rightarrow \mathrm{nsub}(x_1)\mathrm{dobj}(x_2, x_3)$
$\mathrm{dobj}(x_1, x_2 \langle \mathrm{T} \rangle x_3) \rightarrow \mathrm{nsub}(x_1)\mathrm{dobj}(x_2, x_3)$

vs.

$\mathrm{root}(x_1 x_2 x_3, \langle \mathrm{T} \rangle) \rightarrow \mathrm{aux}(x_1, x_2)\mathrm{nsub}(x_3)$
$\mathrm{dobj}(\langle \mathrm{T} \rangle x_1, x_2) \rightarrow \mathrm{nsub}(x_1)\mathrm{det}(x_2)$

String of rule: $\mathrm{A}(x_1 x_2 \langle \mathrm{T} \rangle, x_3) \rightarrow \mathrm{B}(x_1)\mathrm{C}(x_2, x_3) \rightsquigarrow \mathrm{BC}\langle \mathrm{T} \rangle, \mathrm{C}$

Bigram	BC⟨T⟩,C	⟨T⟩,CBCB	Δ
BC	1	1	0
C⟨T⟩	1	0	1
⟨T⟩,	1	1	0
,C	1	1	0
CB	0	2	2

Damerau–Levenshtein distance

- edit distance between two sequences
- minimum number of operations to change one word into the other
- insertions, deletions or substitutions, transposition

Training of the embedding

Gradient descent algorithm

$$\text{Loss} = \sum_{r_1 \in P} \sum_{r_2 \in P} \mu(r_1, r_2) \cdot \text{dist}(e(r_1), e(r_2)) + \lambda(||e(r_1)||_2^2 + ||e(r_2)||_2^2) \rightarrow \min$$

Training of the embedding

Gradient descent algorithm

$$\text{Loss} = \sum_{r_1 \in P} \sum_{r_2 \in P} \mu(r_1, r_2) \cdot \text{dist}(e(r_1), e(r_2)) + \lambda(||e(r_1)||_2^2 + ||e(r_2)||_2^2) \rightarrow \min$$

$\mu \colon P \times P \rightarrow [-1, 1] \ldots$ Similarity measure

Training of the embedding

Gradient descent algorithm

$$\text{Loss} = \sum_{r_1 \in P} \sum_{r_2 \in P} \mu(r_1, r_2) \cdot \text{dist}(e(r_1), e(r_2)) + \lambda(||e(r_1)||_2^2 + ||e(r_2)||_2^2) \rightarrow \min$$

$\mu\colon P \times P \rightarrow [-1, 1]$ ... Similarity measure

$e\colon P \times P \rightarrow \mathbb{R}^n$ ... Embedding function

Training of the embedding

Gradient descent algorithm

$$\text{Loss} = \sum_{r_1 \in P} \sum_{r_2 \in P} \mu(r_1, r_2) \cdot \text{dist}(e(r_1), e(r_2)) + \lambda(||e(r_1)||_2^2 + ||e(r_2)||_2^2) \rightarrow \min$$

$\mu \colon P \times P \rightarrow [-1, 1] \ldots$ Similarity measure

$e \colon P \times P \rightarrow \mathbb{R}^n \ldots$ Embedding function

$|| \cdot ||_2^2 = \sum_{i=1}^{n} (x_i)^2 \ldots$ Square of L2-Norm

Training of the embedding

Gradient descent algorithm

$$\text{Loss} = \sum_{r_1 \in P} \sum_{r_2 \in P} \mu(r_1, r_2) \cdot \text{dist}(e(r_1), e(r_2)) + \lambda(||e(r_1)||_2^2 + ||e(r_2)||_2^2) \rightarrow \min$$

$\mu \colon P \times P \rightarrow [-1, 1]$ ... Similarity measure

$e \colon P \times P \rightarrow \mathbb{R}^n$ ... Embedding function

$||\cdot||_2^2 = \sum_{i=1}^{n} (x_i)^2$ ... Square of L2-Norm

$\lambda \in \mathbb{R}$ ... Regularization factor

Training of the embedding

Gradient descent algorithm

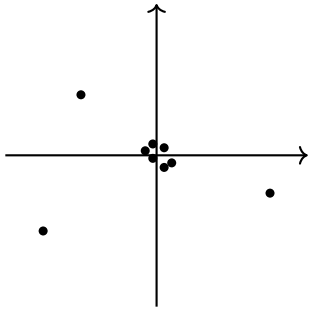
$$\text{Loss} = \sum_{r_1 \in P} \sum_{r_2 \in P} \mu(r_1, r_2) \cdot \text{dist}(e(r_1), e(r_2)) + \lambda(||e(r_1)||_2^2 + ||e(r_2)||_2^2) \rightarrow \min$$

$\mu \colon P \times P \rightarrow [-1, 1]$ ... Similarity measure

$e \colon P \times P \rightarrow \mathbb{R}^n$ ... Embedding function

$|| \cdot ||_2^2 = \sum_{i=1}^{n} (x_i)^2$ ... Square of L2-Norm

$\lambda \in \mathbb{R}$ ... Regularization factor



Training of the embedding

Gradient descent algorithm

$$\text{Loss} = \sum_{r_1 \in P} \sum_{r_2 \in P} (1 - \mu(r_1, r_2) - \text{dist}(e(r_1), e(r_2)))^2 \rightarrow \min$$

$\mu \colon P \times P \rightarrow [0, 1]$ ... Similarity measure

$e \colon P \times P \rightarrow \mathbb{R}^n$ ... Embedding function

$|| \cdot ||_2^2 = \sum_{i=1}^{n} (x_i)^2$ ... Square of L2-Norm

Evaluation - Parse Failure

	$N = 25$	$N = 50$	$N = 75$	$N = 100$
Bigram	66.61/0.00	13.15/0.98	1.33/8.6	0.07/19.63
Trigram	95.76/0.00	10.09/0.12	0.75/1.52	0.38/3.69
Levenshtein	99.31/0.26	90.00/8.84	71.72/26.16	52.56/44.53

- Format: Parse failure / timeout in % (Timeout = 60s)
- Levenshtein is unable to parse most sentences
- Larger N decrease parse failure % but increase timeout %

Evaluation - Time

	$N = 25$	$N = 50$	$N = 75$	$N = 100$
Bigram	42/19	1965/189	8468/422	16725/1815
Trigram	27/15	922/191	3344/540	4923/705
Levenshtein	165/1	5401/3	15691/4	27170/14

- Format: Time average / median in ms
- Time increases with larger N
- Trigram is faster than Bigram
- Median much lower than average

Evaluation - Precision

	$N = 25$	$N = 50$	$N = 75$	$N = 100$
Bigram	76.14/82.00	60.74/67.54	60.44/67.95	61.98/69.76
Trigram	44.46/64.32	8.82/21.51	10.77/23.59	12.49/25.15
Levenshtein	90.79/96.65	90.42/96.64	90.10/97.00	89.01/96.35

- Format: unabled / labeled in %
- Trigram much worse than Bigram
- Labeled and unlabeled precision similar

Summary

- Supertagging: Pre-selection of small number of rules to increase performance
- Neural network BERT can be used
- requires vector encoding of LCFRS-rules
- training embedding based on similarity measures
- of the ones tested, bigram performed the best

Literature

Devlin, J. ; Chang, M.-W. ; Lee, K. ; Toutanova, K. :

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186

Joshi, A. K. ; Srinivas, B. :

Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing.

In: *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*

Kuhlmann, M. ; Satta, G. :

Treebank Grammar Techniques for Non-Projective Dependency Parsing.

In: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Association for Computational Linguistics, 478–486

Nederhof, M. :

Weighted Deductive Parsing and Knuth's Algorithm.

In: *Computational Linguistics* 29 (2003), Nr. 1, S. 135–143