Parsing Beyond Context-Free Grammars: Supertagging with LTAGs

slides to a large extent done by Tatiana Bladier

Laura Kallmeyer
Heinrich-Heine-Universität Düsseldorf

Wintersemester 2021/22

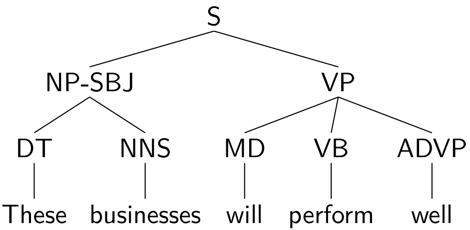
Overview

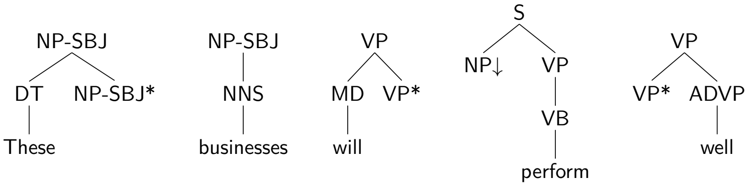
1. Supertagging: Idea
2. Supertagging models

Supertagging: Idea

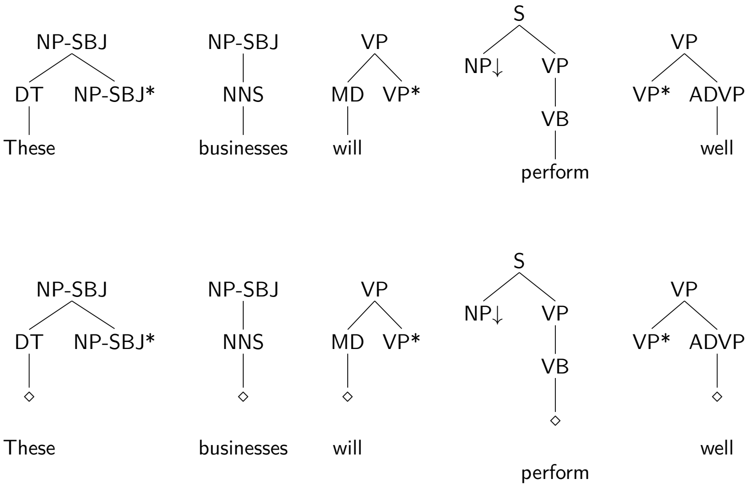
- Supertagging is similar to POS-tagging
- Supertags = super part of speech tags [JS94]
- Computation of a linguistic structure can be localized if lexical items are associated with rich descriptions (supertags)
- Local ambiguity (i.e. the choice of supertags) can be resolved by using **statistical distributions of supertag co-occurences**
  - ⋆ these statistics are collected from a corpus of parses
  - ⋆ for example, from an LTAG-annotated treebank
- These supertag disambiguation results in a representation which is effectively a parse (**an almost parse**)

Supertagging: Extraction of the supertags



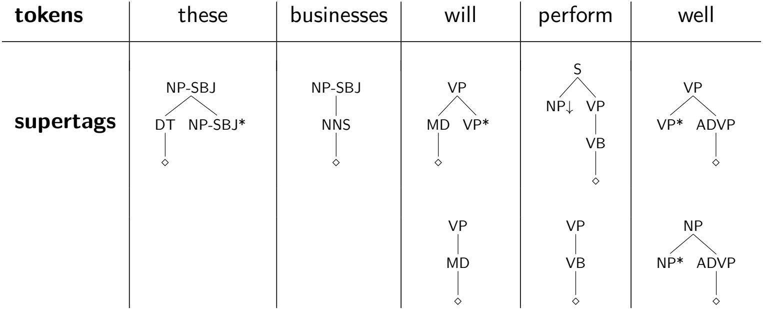


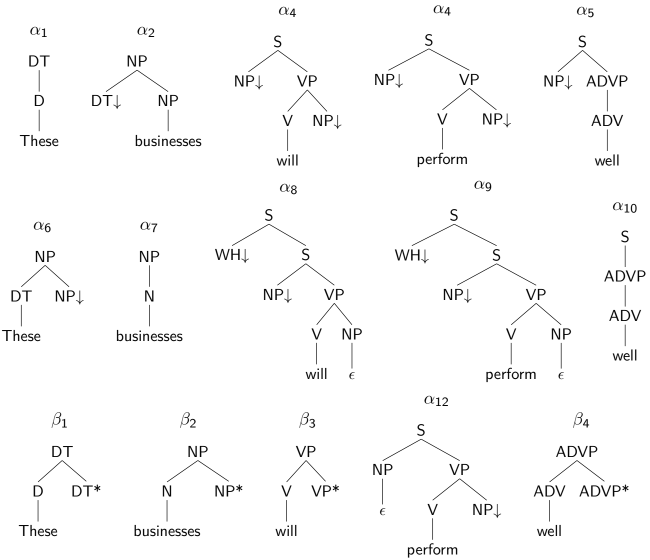
Extraction of the supertags



Creating a lexicon of token-supertags pairs

- N-to-M relation: one token can have several supertags
- One supertag can be attached to several tokens
- Every supertag has an assigned probability (dependent on the context)




α_1
DT
D
These
α_2
NP
DT↓
NP
businesses
α_4
S
NP↓
VP
V
NP↓
will
α_4
S
NP↓
VP
V
NP↓
perform
α_5
S
NP↓
ADVP
ADV
well
α_6
NP
DT
NP↓
These
α_7
NP
N
businesses
α_8
S
WH↓
S
NP↓
VP
V
NP
will
ϵ
α_9
S
WH↓
S
NP↓
VP
V
NP
perform
ϵ
α_{10}
S
ADVP
ADV
well
β_1
DT
D
DT*
These
β_2
NP
N
NP*
businesses
β_3
VP
V
VP*
will
α_{12}
S
NP
VP
ϵ
V
NP↓
perform
β_4
ADVP
ADV
ADVP*
well

Supertagging: idea

Sentence:	These	businesses	will	perform	well
Supertag set:	α_1	α_2	α_3	α_4	α_5
	α_6	α_7	α_8	α_9	α_{10}
	β_1	β_2	β_3	α_{12}	β_4
Final assignement	β_1	α_7	β_3	α_{12}	β_4

Supertagging: statistics

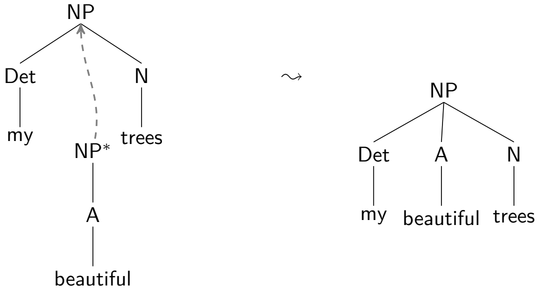
- The number of distinct LTAG supertags extracted from different treebanks is different, but is approximately around 4000 (see the table below)
- Almost the half of all supertags appear just once in an LTAG-annotated corpus

Parameters	**French** French Treebank [BvCSK18]	**German** TiGer Treebank [Kae12]	**English** Penn Treebank [KFM+17]
Distinct supertags	5145	3426	4727
Supertags occur. once	2693	1562	2165
POS tags	13	53	36
Sentences	21550	50000	44168
Avg. sentence length (in tokens)	31.34	17.51	appr. 20
Accuracy	**78.54**	**88.51**	**89.32**

Supertagging statistics with different Treebanks [BvCSK18]

Side note: sister adjunction

Instead of standard adjunction (or maybe in addition), some approaches also use sister adjunction [Kae12, BvCSK18]:



- Sister adjunction produces flat structures (no binarization needed before extracting modifier trees)
- TAG with only substutiton and sister adjunction is weakly equivalent to CFG

What happens if we do not use supertagging?

- Supertagging makes a pre-choice of possible supertags before actual parsing.
- Supertagging reduces the work of the parser.
- If we do not use a supertagging step, the parser's job looks as follows:
  - ⋆ Pick all possible supertags from the big lexicon of tokens for every token in the sentence
    $\Rightarrow$ we might end up with hundreds of possible supertags for every item in the sentence
  - ⋆ Try to figure out the right combination of the supertags to derive a parse of the whole sentence
    $\Rightarrow$ might lead to high computational costs for every sentence – depending on the implemented parsing algorithm
    $\Rightarrow$ Leading, for example, to long reaction times of the parser

N-best supertagging [HH04]

- 1-best supertagging:
  - ⋆ Predict for every token in a sentence just one possible supertag
- n-best supertagging:
  - ⋆ Predict for every token in a sentence a set of most probable supertags (2-best, 3-best, 5-best etc.)
    $\Rightarrow$ increases the accuracy of the supertagger
    $\Rightarrow$ while keeping the number of possible supertags for the disambiguation relatively small:

n-best	**Accuracy German**	**Accuracy French**
1-best	88.51	78.54
2-best	94.37	87.34
3-best	96.08	90.85
5-best	97.45	94.38
7-best	98.03	96.00
10-best	98.52	97.08

Table : N-best supertagging experiments [BvCSK18]

N-best supertagging [HH04]

- In order to find n best sequence hypotheses for a sequence of coded words [HH04]:
  - ⋆ Let $t_1^N = t_1 t_2 \dots t_N$ be a sequence of supertags
  - ⋆ for a sentence $w_1^N = w_1 w_2 \dots w_N$
  - ⋆ then the most probable supertag sequence $\hat{t}_1^N$ is calculated as follows:
    $\hat{t}_1^N = \text{argmax}_{t_1^N} P(t_1^N | w_1^N)$
    $P(t_1^N | w_1^N) \approx \prod_{i=1}^{N} P(t_i | t_{i-2} t_{i-1}) P(w_i | t_i)$

Supertagging models

- *n-gram* model [JS94]
- dependency model [JS94]
- Hidden Markov Model (HMM) based supertagging [HH04]
- Neural supertagging (RNNs) [KFM+17]
- Fine-tuning of contextual language models for supertagging [Sch21]

[Ban97, Fai09, BJ10, BJ99]

N-gram model

- This method is sensitive to the context.
- Contextual dependency probabilities between supertags within a window of n words
- For example, 3-gram model: accuracy of 68% [JS94]
- The probability of x_i is based on the probabilities of $x_{i-(n-1)}, \dots, x_{i-1}$: $P(x_i | x_{i-(n-1)}, \dots, x_{i-1})$

Dependency model

- In the n-gram model, dependencies between supertags beyond the window size are not captured.
- *Dependency model* does not have a pre-defined size of the window
- A supertag is seen as dependent on another supertag if the former substitutes or adjoins into the latter
- Accuracy of 77% [JS94]

Dependency model

Sentence:	These	businesses	will	perform	well
Supertag set:	α_1	α_2	α_3	α_4	α_5
	α_6	α_7	α_8	α_9	α_{10}
	β_1	β_2	β_3	α_{12}	β_4
Final assignement	β_1	α_7	β_3	α_{12}	β_4

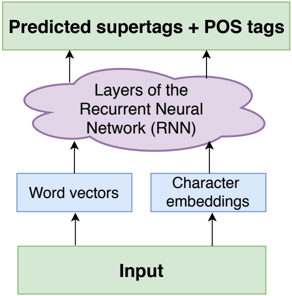
Dependency model

- A tree α_4, anchored by a verb (V), has a left and a right dependent, and the first word to the left (-1) with the tree α_2 is dependent on the current word.
- The algorithm proceeds to satisfy dependency requirements of α_4 in both directions
- It picks a dependency data entry and proceeds to set up a path with the first word to the left that has the dependent supertag label (α_2)
- If the first word satisfies this requirement, an arc is set up between α_2 and α_4
- A successful supertag sequence is one which assigns a supertag to each position and each supertag has all of its dependents and this sequence has the highest probability

POS, Supertag	Direction of Dependent Supertag	Dependent Supertag	Ordinal Position	Prob.
(D, α_1)	( )	-	-	-
(N, α_2)	($-$)	α_1	-1	0.90
(V, α_3)	( )	-	-	-
(V, α_4)	($-$,$+$)	α_8	-2	0.700
(V, α_4)	($-$,$+$)	α_2	-1	0.300
(V, α_4)	($-$,$+$)	α_8	$+1$	0.300
(ADV, α_2)	( )	-	-	-

HMM-based and RNN-based supertagging models

- Supertagging as a task of a sequence labeling
- Hidden Markov Models or Recurrent Neural Networks are able to capture the dependencies between the supertags [HH04, KFM+17]
- The tokens are presented in small batches (called *windows*), for example 5 tokens at a time ($window\ size = 5$)
- Accuracy of about 90 %



"the cat"

word vectors (100 dimensions):

[[-0.075408, . . . , -0.20341, 0.17471],
[-0.3058, . . . , 0.41226, 0.047526]]

character embeddings:

[[22, 8, 5], [3, 1, 22]]

supertags:

[(NP* (DET ⋄)), (NP (N ⋄))]

References I

[Ban97] Srinivas Bangalore.
Complexity of lexical descriptions and its relevance to partial parsing.
IRCS Technical Reports Series, page 82, 1997.

[BJ99] Srinivas Bangalore and Aravind K Joshi.
Supertagging: An approach to almost parsing.
Computational linguistics, 25(2):237–265, 1999.

[BJ10] Srinivas Bangalore and Aravind K Joshi.
Supertagging: Using Complex Lexical Descriptions in Natural Language Processing.
The MIT Press, 2010.

[BvCSK18] Tatiana Bladier, Andreas van Cranenburgh, Younes Samih, and Laura Kallmeyer.
German and french neural supertagging experiments for ltag parsing (to appear).
ACL 2018 Student Research Workshop, Melbourne, Australia, 2018.

[Fai09] Heshaam Faili.
From partial toward full parsing.
In *Proceedings of the International Conference RANLP-2009*, pages 71–75, 2009.

[HH04] Saša Hasan and Karin Harbusch.
N-best hidden markov model supertagging to improve typing on an ambiguous keyboard.
In *Proceedings of the 7th International Workshop on Tree Adjoining Grammar and Related Formalisms*, pages 24–31, 2004.

[JS94] Aravind K Joshi and Bangalore Srinivas.
Disambiguation of super parts of speech (or supertags): Almost parsing.
In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 154–160. Association for Computational Linguistics, 1994.

References II

[Kae12] Miriam Kaeshammer.
A German treebank and lexicon for tree-adjoining grammars.
PhD thesis, Master's thesis, Universitat des Saarlandes, Saarlandes, Germany, 2012.

[KFM+17] Jungo Kasai, Robert Frank, Tom McCoy, Owen Rambow, and Alexis Nasr.
Tag parsing with neural networks and vector representations of supertags.
In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, 2017.

[Sch21] Svetlana Schmidt.
Transformers-based supertagging model for statistical parsing of tree wrapping grammars, 2021.
BA Thesis, Heinrich Heine Universität Düsseldorf, Germany.