New supertaggers for the ERG

for the DELPH-IN summit

Olga Zamaraeva & Carlos Gómez-Rodríguez
Department of Informatics/CITIC, Universidade da Coruña

July 17 2022

Introduction: Supertagging

- Supertagging is like POS tagging:
  - Sequence-to-sequence statistical problem
    - input seq: sentence or text
    - output seq: tags corresponding to each token*
  - usually with more fine-grained tags
  - e.g. lexical types in HPSG grammars

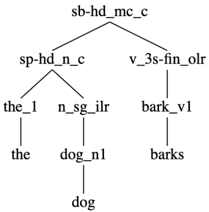
Supertagging is useful for e.g.>0

- word sense disambiguation
- parse ranking
- **improving parsing speed**

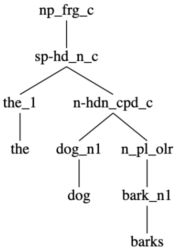
Supertagging for parsing speed

- Without tagging:
  - many tokens can be mapped to more than one lexical entry/type
- Tagging helps eliminate unlikely possibilities
  - less things for the parser to go through

1



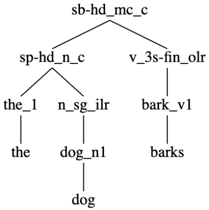
2



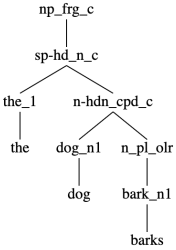
Supertagging accuracy is important

- Supertagging can mean discarding all possibilities but one
- If a wrong lexical type is predicted:
  - the chances of getting the parse right are 0

1



2



Prior work on supertagging for HPSG

model	grammar	training tok	tagset size	speed-up factor
N-gram (Prins and van Noord 2004)	Alpino (Dutch)	24 mln	1365	2
HMM (Blunsom 2007)	ERG (English)	113K	615	8.5
MEMM (Dridan 2009)	ERG (English)	158K	676	12

- Dridan (2009):
  - 92% accuracy on in-domain data
  - 74.6% out of domain (up to 80.8% with additional training data)
- Recent **related** work on CCG (Liu et al. 2021):
  - 95.5% accuracy in domain
  - 81% and 92.4% on two out-of-domain datasets

New experiments with ERG 2020

- Supertagging (no ubertagging)
- Single tag accuracy (top 1)
- Tagset: 1127
- Not yet integrated into any parser

dataset[11]	description	sent	tok	train tok[12]	MaxEnt	SVM	neural
cb	technical essay	769	17,244	0	88.96	89.53	**91.94**
ecpr	e-commerce	1207	11,550	24,934	91.80	91.99	**95.09**
jh*,tg*,ps*, ron*	travel brochures	2102	34,098	147,166	90.45	91.21	**95.44**
petet	textual entailment	602	7135	1578	92.88	95.31	**96.93**
vm32	phone customer service	1034	8730	86,630	93.57	94.29	**95.62**
ws213-214	Wikipedia	1613	29,697	161,623	91.31	92.02	**93.66**
wsj23	Wall Street Journal	1000[13]	22,987	959,709	94.27	94.72	**96.05**
all	all test sets as one	8,327	131,441	1,381,645	91.57	92.28	**94.46**
all	average	8,327	131,441	1,381,645	91.89	92.72	**94.96**
speed (sen/sec)	average	8,327	131,441	1,381,645	1024	**7414**	125

Neural model details

- NCRF++ library (Yang and Zhang 2018)
  - fast for POS but not CCG/HPSG supertagging
- almost out of the box:
  - unknown label handling added

Parameter	value	default/tuned
lstm layers	2	tuned
hidden dim.	800	tuned
word embeddings	glove840B	pretrained
word emb. dim.	300	N/A
char emb. dim.	50	tuned
momentum	0	default
dropout	0.5	default
l2	1^{-8}	default

Conclusion

- We can have more accurate supertaggers for ERG

Conclusion

- We can have more accurate supertaggers for ERG
- scikit-learn SVM (Pedregosa et al. 2011) might be the practical model to integrate into ACE for sentence-by-sentence processing

Conclusion

- We can have more accurate supertaggers for ERG
- scikit-learn SVM (Pedregosa et al. 2011) might be the practical model to integrate into ACE for sentence-by-sentence processing
  - Neural models even more accurate but need faster implementation

Conclusion

- We can have more accurate supertaggers for ERG
- scikit-learn SVM (Pedregosa et al. 2011) might be the practical model to integrate into ACE for sentence-by-sentence processing
  - Neural models even more accurate but need faster implementation
  - Any takers? :)

References

Philip Blunsom. 2007. *Structured classification for multilingual natural language processing*. PhD thesis, University of Melbourne.

Rebecca Dridan. 2009. *Using lexical statistics to improve HPSG parsing*. PhD thesis, University of Saarland.

Yufang Liu, Tao Ji, Yuanbin Wu, and Man Lan. 2021. Generating ccg categories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13443–13451.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

RP Prins and GJM van Noord. 2004. Reinforcing parser preferences through tagging. *Traitement Automatique des Langues*, 3:121–139.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. URL http://aclweb.org/anthology/P18-4013.