Stacking or Supertagging for Dependency Parsing
What's the Difference?

Agnieszka Faleńska, Anders Björkelund, Özlem Çetinoğlu and Wolfgang Seeker

Universität Stuttgart

Section 1

Introduction

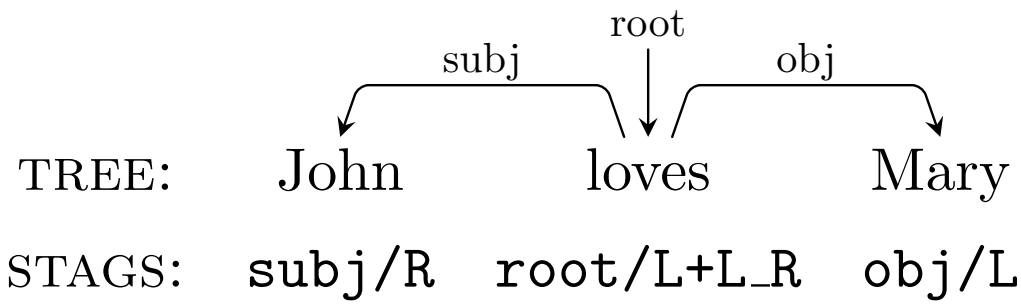
Introduction - the title

Stacking or Supertagging for Dependency Parsing – What's the Difference?

Supertagging

Supertags - labels for tokens encoding syntactic information

Example from [Ouchi et al., 2014]:

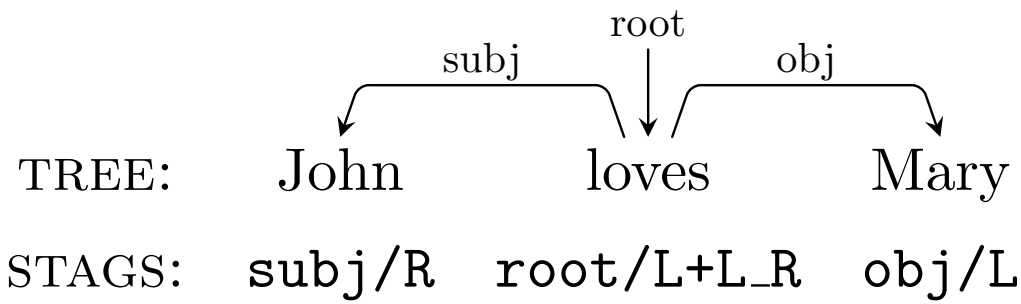


Supertags are usually predicted by sequence labelers or classifiers.

Supertagging

Supertags - labels for tokens encoding syntactic information

Example from [Ouchi et al., 2014]:



Supertags are usually predicted by sequence labelers or classifiers.

Supertagging - background

- Joshi and Bangalore [1994] - elementary structures associated with a lexical item
- Bangalore and Joshi [1999]
  - a supertagger assigns supertags to each word of a sentence
  - a parser combines these structures into a full parse
  - they speed up the parser
- Clark and Curran [2004] - Combinatory Categorial Grammars
- Foth et al. [2006] - dependency parsing context
  - supertags as soft constraints in rule-based parser

Traditional use:

- reduce the search space
- score possible analyses

Supertagging - background

- Joshi and Bangalore [1994] - elementary structures associated with a lexical item
- Bangalore and Joshi [1999]
  - a supertagger assigns supertags to each word of a sentence
  - a parser combines these structures into a full parse
  - they speed up the parser
- Clark and Curran [2004] - Combinatory Categorial Grammars
- Foth et al. [2006] - dependency parsing context
  - supertags as soft constraints in rule-based parser

Traditional use:

- reduce the search space
- score possible analyses

Supertagging - background

- Joshi and Bangalore [1994] - elementary structures associated with a lexical item
- Bangalore and Joshi [1999]
  - a supertagger assigns supertags to each word of a sentence
  - a parser combines these structures into a full parse
  - they speed up the parser
- Clark and Curran [2004] - Combinatory Categorial Grammars
- Foth et al. [2006] - dependency parsing context
  - supertags as soft constraints in rule-based parser

Traditional use:

- reduce the search space
- score possible analyses

Supertagging - background

- Joshi and Bangalore [1994] - elementary structures associated with a lexical item
- Bangalore and Joshi [1999]
  - a supertagger assigns supertags to each word of a sentence
  - a parser combines these structures into a full parse
  - they speed up the parser
- Clark and Curran [2004] - Combinatory Categorial Grammars
- Foth et al. [2006] - dependency parsing context
  - supertags as soft constraints in rule-based parser

Traditional use:

- reduce the search space
- score possible analyses

Supertagging - background

- Joshi and Bangalore [1994] - elementary structures associated with a lexical item
- Bangalore and Joshi [1999]
  - a supertagger assigns supertags to each word of a sentence
  - a parser combines these structures into a full parse
  - they speed up the parser
- Clark and Curran [2004] - Combinatory Categorial Grammars
- Foth et al. [2006] - dependency parsing context
  - supertags as soft constraints in rule-based parser

Traditional use:

- reduce the search space
- score possible analyses

Supertagging - background (2)

Recently - a method to provide syntactic information to the feature model of a statistical dependency parser:

- Ambati et al. [2013, 2014] - CCG categories improve a dependency parser (English, Hindi)
- Ouchi et al. [2014] - supertags extracted from a dependency treebank (English)
- Björkelund et al. [2014] - nine other languages

In this presentation - supertagging as a way of incorporating syntactic features to dependency parsers.

Supertagging - background (2)

Recently - a method to provide syntactic information to the feature model of a statistical dependency parser:

- Ambati et al. [2013, 2014] - CCG categories improve a dependency parser (English, Hindi)
- Ouchi et al. [2014] - supertags extracted from a dependency treebank (English)
- Björkelund et al. [2014] - nine other languages

In this presentation - supertagging as a way of incorporating syntactic features to dependency parsers.

Supertagging - background (2)

Recently - a method to provide syntactic information to the feature model of a statistical dependency parser:

- Ambati et al. [2013, 2014] - CCG categories improve a dependency parser (English, Hindi)
- Ouchi et al. [2014] - supertags extracted from a dependency treebank (English)
- Björkelund et al. [2014] - nine other languages

In this presentation - supertagging as a way of incorporating syntactic features to dependency parsers.

Stacking

Stacking - one parser uses the output of the second parser as features (for example, whether a particular arc was predicted)

- introduced by Nivre and McDonald [2008]
- Martins et al. [2008] - extend feature set with non-local information
- Surdeanu and Manning [2010] - the diversity of the parsing algorithms is an important factor while stacking

What's the Difference?

- two ways of improving a statistical dependency parser
- two separate ideas successful independently

- intuitively - they have much in common
- hypothesis: supertagging is a form of stacking
- questions:
  - does stacking give higher improvements than supertagging?
  - what is the best/fastest way to realize those methods?
  - is there any benefit from combining them?

What's the Difference?

- two ways of improving a statistical dependency parser
- two separate ideas successful independently

- intuitively - they have much in common
- hypothesis: supertagging is a form of stacking
- questions:
  - does stacking give higher improvements than supertagging?
  - what is the best/fastest way to realize those methods?
  - is there any benefit from combining them?

What's the Difference?

- two ways of improving a statistical dependency parser
- two separate ideas successful independently

- intuitively - they have much in common
- hypothesis: supertagging is a form of stacking
- questions:
  - does stacking give higher improvements than supertagging?
  - what is the best/fastest way to realize those methods?
  - is there any benefit from combining them?

What's the Difference?

- two ways of improving a statistical dependency parser
- two separate ideas successful independently

- intuitively - they have much in common
- hypothesis: supertagging is a form of stacking
- questions:
  - does stacking give higher improvements than supertagging?
  - what is the best/fastest way to realize those methods?
  - is there any benefit from combining them?

Three groups of experiments

1. Comparing supertagging and stacking
   - does stacking give higher improvements than supertagging?
2. Supertagging without parsers
   - what is the best/fastest way to realize those methods?
3. Combining supertagging and stacking
   - is there any benefit from combining them?

Three groups of experiments

1. Comparing supertagging and stacking
   - (1) accuracy
   - (2) oracle experiments
   - (3) self-application
2. Supertagging without parsers
   - (4) a CRF sequence labeller
   - (5) a greedy transition-based parser
   - (6) out-of-domain application
3. Combining supertagging and stacking
   - (7) combining the same source
   - (8) combining different sources

Three groups of experiments

1. Comparing supertagging and stacking
   - (1) accuracy
   - (2) oracle experiments
   - (3) self-application
2. Supertagging without parsers
   - (4) a CRF sequence labeller
   - (5) a greedy transition-based parser
   - (6) out-of-domain application
3. Combining supertagging and stacking
   - (7) combining the same source
   - (8) combining different sources

Section 2

Experimental Setup

Data Sets and Preprocessing

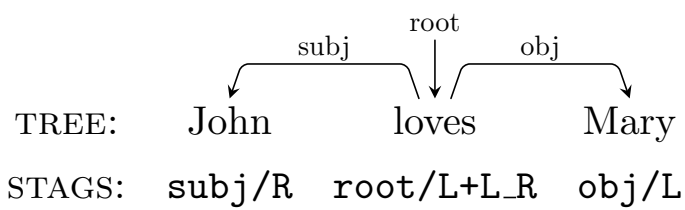
- 10 languages:
  - the SPMRL 2014 Shared Task's data sets:
    - Arabic
    - Basque
    - French
    - Hebrew
    - German
    - Hungarian
    - Korean
    - Polish
    - Swedish
  - \+ English Penn Treebank converted to Stanford Dependencies
- automatically predicted preprocessing
  - POS tags and morphological features by MarMoT [Müller et al., 2013]
  - the mate-tools for lemmatization

Supertag Design

Multiple options for supertags model design:

- Foth et al. [2006] - richer supertags improve parser's accuracy (but are harder to predict)
- Ouchi et al. [2014] - difference between models on tests sets not significant

Model 1 from [Ouchi et al., 2014]:



Notation

- $STACK^{y}_{x}$ - y uses output of x in stacking
- $STAG^{y}_{x}$ - y uses supertags provided by x
- x - Level 0 tool
  y - Level 1 tool

Parsers

- the transition-based parser *TB*
  - an in-house implementation using the arc-standard decoding algorithm with a swap transition [Nivre, 2009]
- the graph-based parser *GB*
  - TurboParser version 2.0.1
- in this presentation - all plots for the graph-based parser

Parsers

- the transition-based parser *TB*
  - an in-house implementation using the arc-standard decoding algorithm with a swap transition [Nivre, 2009]
- the graph-based parser *GB*
  - TurboParser version 2.0.1
- in this presentation - all plots for the graph-based parser

Feature Models

- a simpler feature set is more useful for a comparison
- the supertag features mimic the information provided by stacking (to the best extent possible)
- *GB* example (h, d - the head and the dependent):
  - stacking: $\text{head}(d) = h$
  - supertagging:
    - $\text{hasL}(h) \oplus \text{hdir}(d)$
    - $\text{hasR}(h) \oplus \text{hdir}(d)$

Feature Models

- a simpler feature set is more useful for a comparison
- the supertag features mimic the information provided by stacking (to the best extent possible)
- *GB* example (h, d - the head and the dependent):
  - stacking: $\text{head}(d) = h$
  - supertagging:
    - $\text{hasL}(h) \oplus \text{hdir}(d)$
    - $\text{hasR}(h) \oplus \text{hdir}(d)$

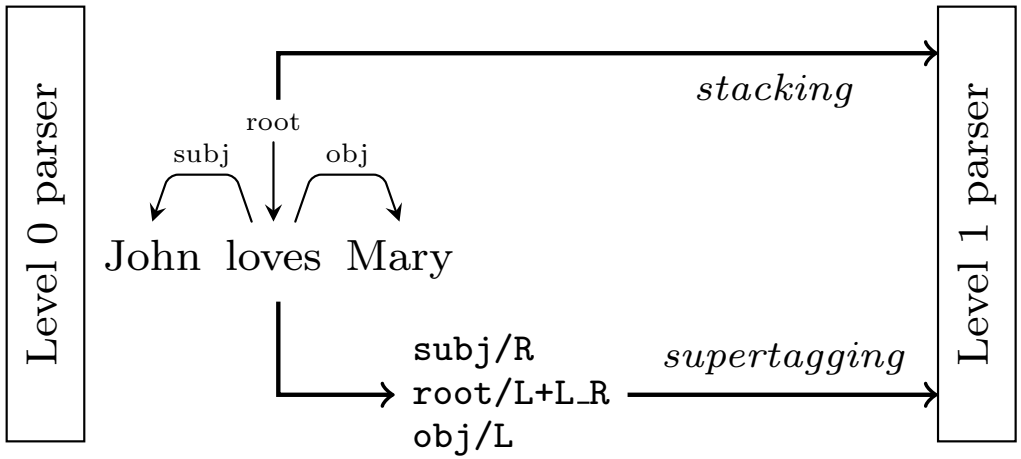
Feature Models

- a simpler feature set is more useful for a comparison
- the supertag features mimic the information provided by stacking (to the best extent possible)
- *GB* example (h, d - the head and the dependent):
  - stacking: $\text{head}(d) = h$
  - supertagging:
    - $\text{hasL}(h) \oplus \text{hdir}(d)$
    - $\text{hasR}(h) \oplus \text{hdir}(d)$

Section 3

Experiments

Comparing Supertagging and Stacking

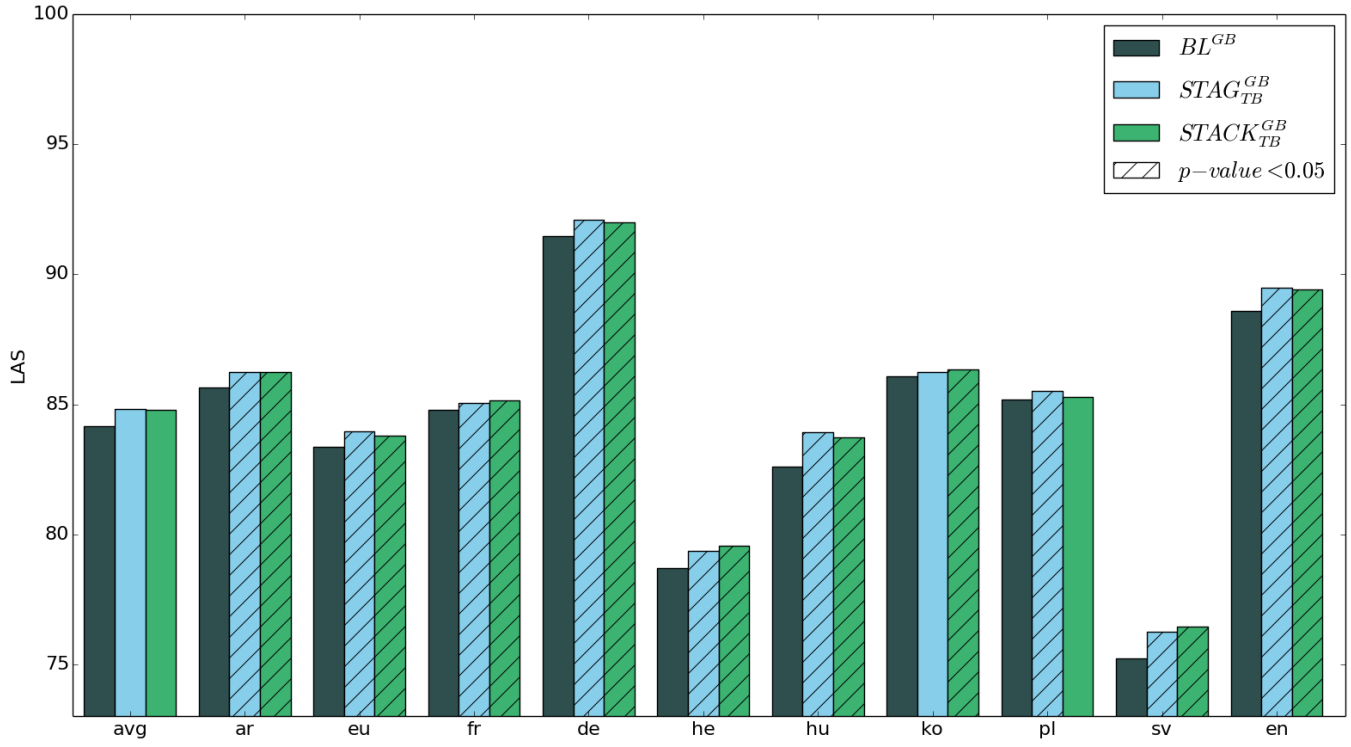


- Level 0 tool is a parser
- focusing on the means by which the information is given to the Level 1 parser

Experiment (1) - supertagging and stacking accuracy

- Purpose - to convince ourselves that both strategies improve over the baseline.
- The baseline setting (BL) - the parser is run without any additional information.

Experiment (1) - the graph-based parser



- averages:
  - BL^{GB} 84.16
  - $STAG^{GB}_{TB}$ 84.81
  - $STACK^{GB}_{TB}$ 84.79
- significance testing - Wilcoxon signed-rank test

Experiment (1) - conclusions

- results confirm the previous findings:
  - supertagging - [Ouchi et al., 2014], [Ambati et al., 2014]
  - stacking - [Nivre and McDonald, 2008], [Martins et al., 2008]

- both methods improve the accuracies to the same extent
- the improvements are similar but they might still come about in different ways

Experiment (1) - conclusions

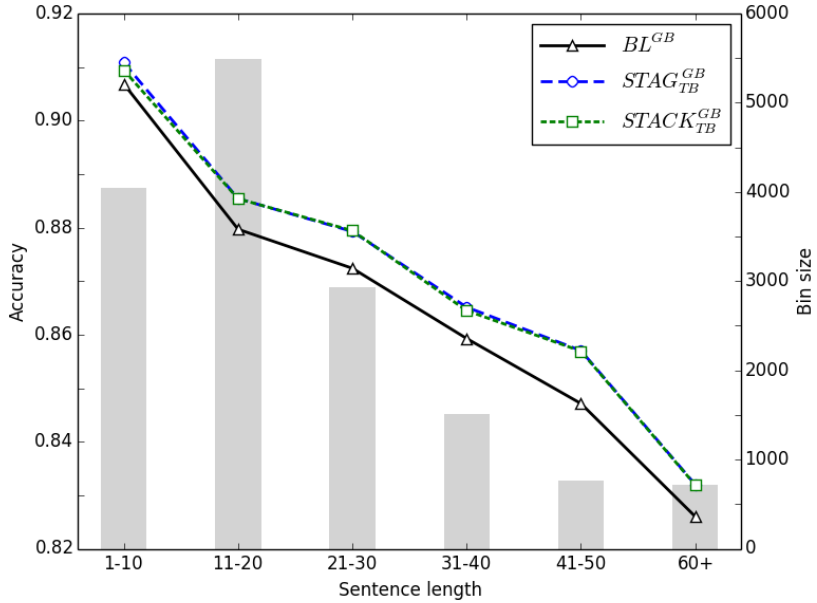
- results confirm the previous findings:
  - supertagging - [Ouchi et al., 2014], [Ambati et al., 2014]
  - stacking - [Nivre and McDonald, 2008], [Martins et al., 2008]
- both methods improve the accuracies to the same extent
- the improvements are similar but they might still come about in different ways

Experiment (1) - conclusions

- results confirm the previous findings:
  - supertagging - [Ouchi et al., 2014], [Ambati et al., 2014]
  - stacking - [Nivre and McDonald, 2008], [Martins et al., 2008]

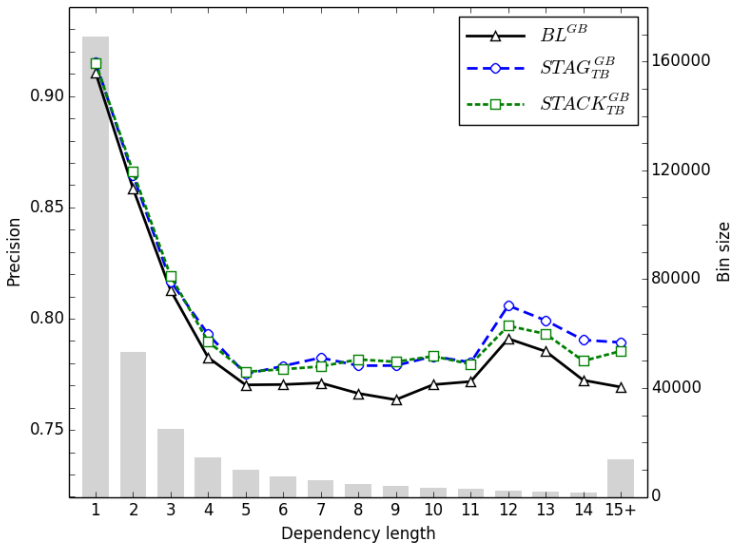
- both methods improve the accuracies to the same extent
- the improvements are similar but they might still come about in different ways

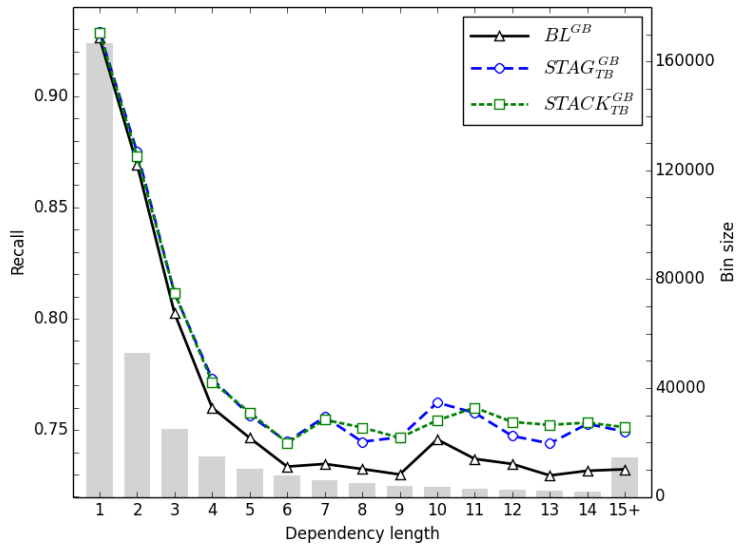
Experiment (1) - in-depth analysis (graph-based parser)



- bins of size 10
- both systems show a consistent improvement over the baseline
- the curves of the stacked and supertagged systems are mostly parallel and close to each other

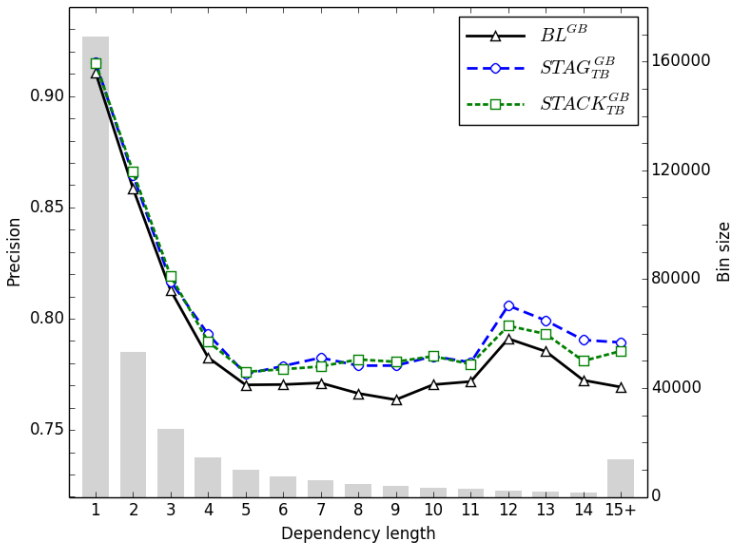
Experiment (1) - in-depth analysis (graph-based parser)

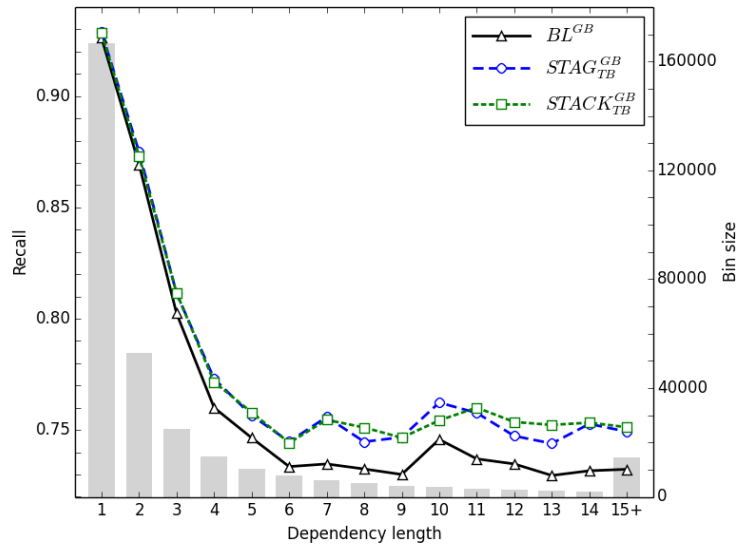




- the improvements are not restricted to sentences or arcs of particular lengths
- Conclusion: both methods are indeed doing the same thing

Experiment (1) - in-depth analysis (graph-based parser)





- the improvements are not restricted to sentences or arcs of particular lengths
- Conclusion: both methods are indeed doing the same thing

Supertagging Without Parsers

Purpose - what is the best way to realize supertagging and stacking?

- most previous work predicts supertags using classifiers or sequence models

Options:

- regular parser (*GB*, *TB*)
- sequence labeler - MarMoT (*SL*)
- fast greedy arc-standard parser (*GTB*)
  - on Arabic 18 times faster than *SL*

Supertagging Without Parsers

Purpose - what is the best way to realize supertagging and stacking?

- most previous work predicts supertags using classifiers or sequence models

Options:

- regular parser (*GB*, *TB*)
- sequence labeler - MarMoT (*SL*)
- fast greedy arc-standard parser (*GTB*)
  - on Arabic 18 times faster than *SL*

Supertagging Without Parsers

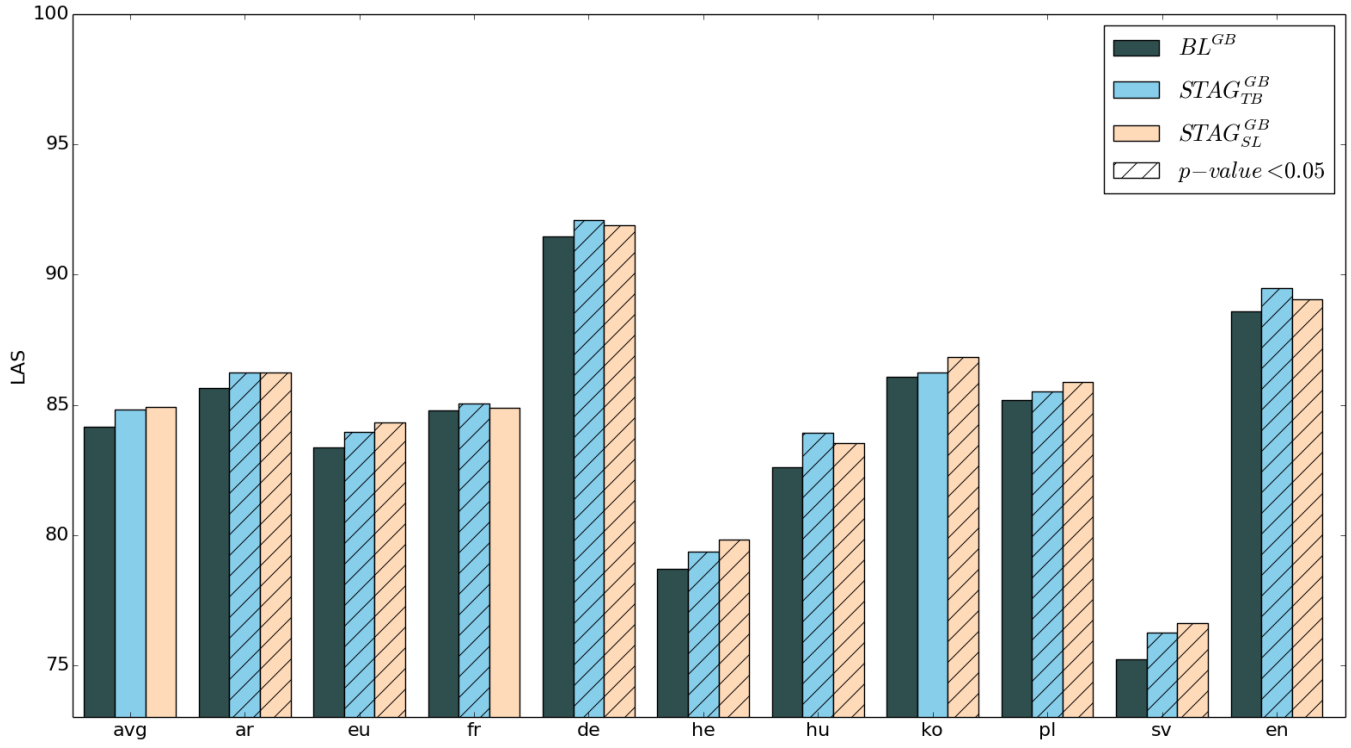
Purpose - what is the best way to realize supertagging and stacking?

- most previous work predicts supertags using classifiers or sequence models

Options:

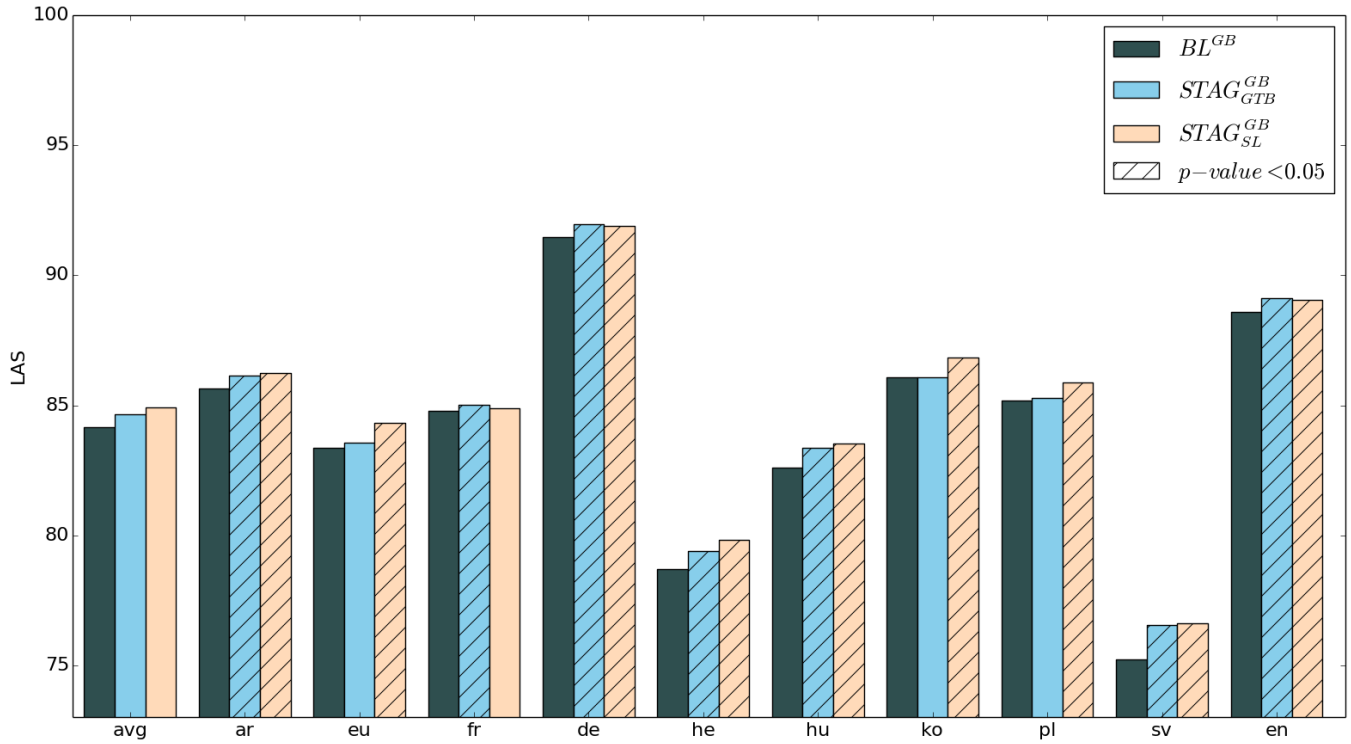
- regular parser (*GB*, *TB*)
- sequence labeler - MarMoT (*SL*)
- fast greedy arc-standard parser (*GTB*)
  - on Arabic 18 times faster than *SL*

Experiment (4) - *TB* v. *SL* (graph-based parser)



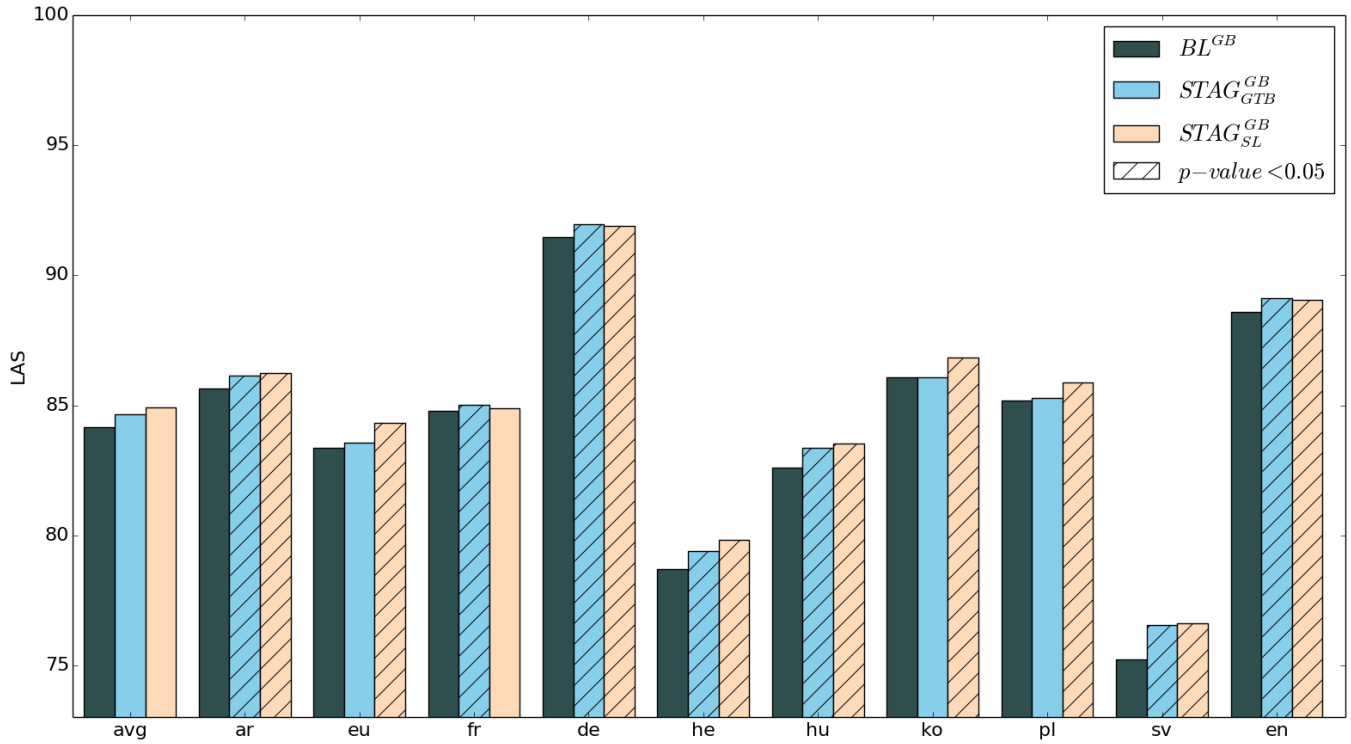
- *SL* is better than the baseline
- on average *SL* is as good as a regular parser
- is *SL* more useful? it depends on the dataset

Experiment (5) - *SL* v. *GTB* (graph-based parser)



- *GTB* slightly behind *SL*
- Conclusion: sequence labelers can be replaced by greedy parsers

Experiment (5) - *SL* v. *GTB* (graph-based parser)

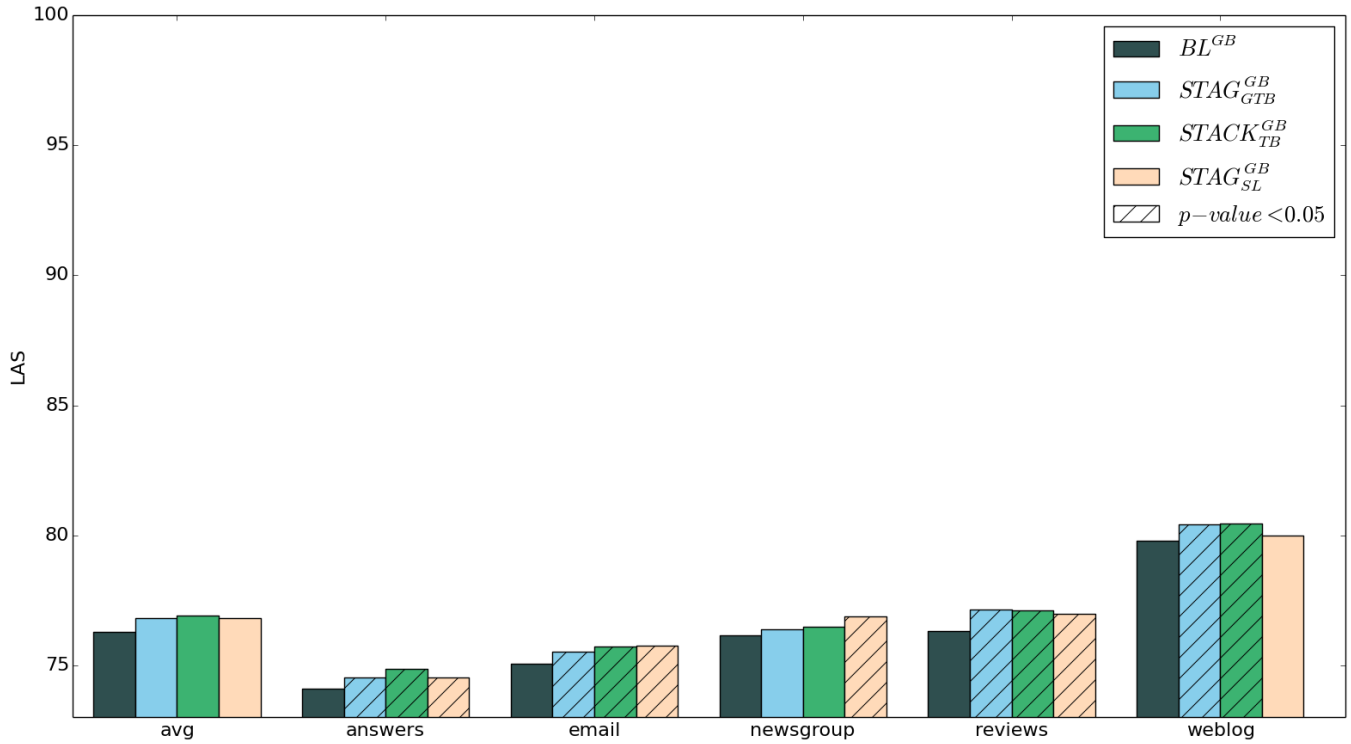


- *GTB* slightly behind *SL*
- Conclusion: sequence labelers can be replaced by greedy parsers

Experiment (6) - out-of-domain application

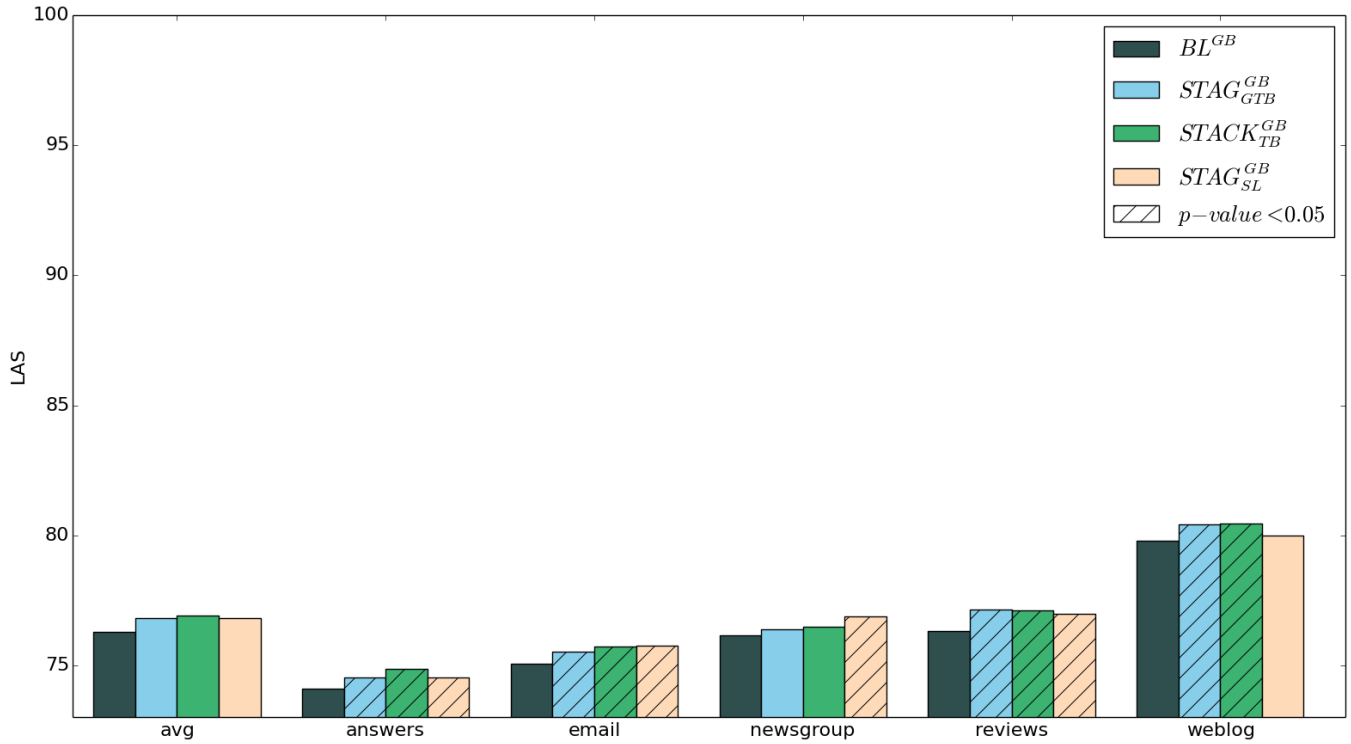
- having fast predictors suggests an application where speed matters
- example - a web data - will the possitive effects propagate into this setting?
- the English Web Treebank [Bies et al., 2012] converted to Stanford Dependency format

Experiment (6) - graph-based parser



- consistent improvements on the five genres
- Conclusion: stagging and stacking are both good methods to improve parsing accuracies when parsing out-of-domain data

Experiment (6) - graph-based parser



- consistent improvements on the five genres
- Conclusion: stagging and stacking are both good methods to improve parsing accuracies when parsing out-of-domain data

Section 4

Conclusions

Conclusions

- a broad range of experiments to compare supertagging with stacking

- conclusions covered by this presentation:
  - supertagging as defined by [Ouchi et al., 2014] is a form of stacking
  - sequence labelers can be replaced by greedy parsers in supertagging
  - supertagging and stacking can improve parsing also in out-of-domain setting

Conclusions

- a broad range of experiments to compare supertagging with stacking

- conclusions covered by this presentation:
  - supertagging as defined by [Ouchi et al., 2014] is a form of stacking
  - sequence labelers can be replaced by greedy parsers in supertagging
  - supertagging and stacking can improve parsing also in out-of-domain setting

Conclusions

- a broad range of experiments to compare supertagging with stacking

- conclusions covered by this presentation:
  - supertagging as defined by [Ouchi et al., 2014] is a form of stacking
  - sequence labelers can be replaced by greedy parsers in supertagging
  - supertagging and stacking can improve parsing also in out-of-domain setting

Conclusions

- a broad range of experiments to compare supertagging with stacking

- conclusions covered by this presentation:
  - supertagging as defined by [Ouchi et al., 2014] is a form of stacking
  - sequence labelers can be replaced by greedy parsers in supertagging
  - supertagging and stacking can improve parsing also in out-of-domain setting

Conclusions

- other conclusions covered by the paper:
  - the intuitive advantage of trees over supertags has no impact in practice (both in realistic and gold scenarios)
  - self-training does not work - neither for stacking nor supertagging
  - combining stacking and supertagging gives improvements only if different tools are used

Conclusions

- other conclusions covered by the paper:
  - the intuitive advantage of trees over supertags has no impact in practice (both in realistic and gold scenarios)
  - self-training does not work - neither for stacking nor supertagging
  - combining stacking and supertagging gives improvements only if different tools are used

Conclusions

- other conclusions covered by the paper:
  - the intuitive advantage of trees over supertags has no impact in practice (both in realistic and gold scenarios)
  - self-training does not work - neither for stacking nor supertagging
  - combining stacking and supertagging gives improvements only if different tools are used

Thank you

Ambati, B. R., Deoskar, T., and Steedman, M. (2013). Using CCG categories to improve Hindi dependency parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 604–609, Sofia, Bulgaria. Association for Computational Linguistics.

Ambati, B. R., Deoskar, T., and Steedman, M. (2014). Improving Dependency Parsers using Combinatory Categorial Grammar. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 159–163, Gothenburg, Sweden. Association for Computational Linguistics.

Bangalore, S. and Joshi, A. K. (1999). Supertagging: An Approach to Almost Parsing. *Computational Linguistics*, 25(2):237–265.

Bies, A., Mott, J., Warner, C., and Kulick, S. (2012). English Web Treebank LDC2012T13.

Björkelund, A., Özlem Çetinoğlu, Faleńska, A., Farkas, R., Müller, T., Seeker, W., and Szántó, Z. (2014). The IMS-Wrocław-Szeged-CIS entry at the SPMRL 2014 Shared Task: Reranking and Morphosyntax meet Unlabeled Data. In *Notes of the SPMRL 2014 Shared Task on Parsing Morphologically-Rich Languages*, Dublin, Ireland.

Clark, S. and Curran, J. R. (2004). The Importance of Supertagging for Wide-coverage CCG Parsing. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Foth, K. A., By, T., and Menzel, W. (2006). Guiding a Constraint Dependency Parser with Supertags. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 289–296, Sydney, Australia. Association for Computational Linguistics.

Joshi, A. K. and Bangalore, S. (1994). Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, pages 154–160, Stroudsburg, PA, USA. Association for Computational Linguistics.

Martins, A. F. T., Das, D., Smith, N. A., and Xing, E. P. (2008). Stacking Dependency Parsers. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 157–166, Honolulu, Hawaii. Association for Computational Linguistics.

Müller, T., Schmid, H., and Schütze, H. (2013). Efficient Higher-Order CRFs for Morphological Tagging. In *In Proceedings of EMNLP*.

Nivre, J. (2009). Non-Projective Dependency Parsing in Expected Linear Time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 351–359, Suntec, Singapore. Association for Computational Linguistics.

Nivre, J. and McDonald, R. (2008). Integrating Graph-Based and Transition-Based Dependency Parsers. In *Proceedings of ACL-08: HLT*, pages 950–958, Columbus, Ohio. Association for Computational Linguistics.

Ouchi, H., Duh, K., and Matsumoto, Y. (2014). Improving Dependency Parsers with Supertags. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 154–158, Gothenburg, Sweden. Association for Computational Linguistics.

Surdeanu, M. and Manning, C. D. (2010). Ensemble Models for Dependency Parsing: Cheap and Good? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 649–652, Los Angeles, California. Association for Computational Linguistics.