Parsing Italian texts together is better than parsing them alone!

Oronzo Antonelli
DISI, University of Bologna, Italy
antonelli.oronzo@gmail.it

Fabio Tamburini
FICLIT, University of Bologna, Italy
fabio.tamburini@unibo.it

Abstract

English. In this paper we present a work aimed at testing the most advanced, state-of-the-art syntactic parsers based on deep neural networks (DNN) on Italian. We made a set of experiments by using the Universal Dependencies benchmarks and propose a new solution based on ensemble systems obtaining very good performances.

Italiano. *In questo contributo presentiamo alcuni esperimenti volti a verificare le prestazioni dei più avanzati parser sintattici sull'italiano utilizzando i* treebank *disponibili nell'ambito delle* Universal Dependencies. *Proponiamo inoltre un nuovo sistema basato sull'*ensemble parsing *che ha mostrato ottime prestazioni.*

1 Introduction

Syntactic parsing of morphologically rich languages like Italian often poses a number of hard challenges. Various works applied different kinds of freely available parsers on Italian training them using different resources and different methods for comparing their results (Lavelli, 2014; Alicante et al., 2015; Lavelli, 2016) and gather a clear picture of the syntactic parsing task performances for the Italian language. In this direction seems relevant to cite the EVALITA[1] periodic campaigns for the evaluation of constituency and dependency parsers devoted to the syntactic analysis of Italian (Bosco and Mazzei, 2011; Bosco et al., 2014).

Other studies regarding the syntactic parsing of Italian tried to enhance the parsing performances by building some kind of *ensemble systems* (Lavelli, 2013; Mazzei, 2015).

By looking at the cited papers we can observe that they evaluated the state-of-the-art parsers before the "neural net revolution" not including the last improvements proposed by new research studies.

The goal of this paper is twofold: first, we would like to test the effectiveness of parsers based on the newly-proposed technologies, mainly deep neural networks, on Italian, and, second, we would like to propose an ensemble system able to further improve the neural parsers performances when parsing Italian texts.

2 The Neural Parsers

We considered nine state of the art parsers representing a wide range of contemporary approaches to dependency parsing whose architectures are based on neural network models (see Table 1). We set-up each parser using the data from the Italian Universal Dependencies (Nivre et al., 2016) treebank, UD Italian 2.1 (general texts) and UD Italian PoSTWITA 2.2 (tweets). For all parsers, we used the default settings for training, following the recommendation of the developers.

In Chen and Manning (2014) dense features are used to learn representations of words, tags and labels using a neural network classifier in order to take parsing decisions within a transition-based greedy model. To address some limitations, in Andor et al. (2016) the authors augmented the parser model with a beam search and a conditional random field loss objective. The work of Ballesteros et al. (2015) extends the parser defined in Dyer et al. (2015) introducing character-level representation of words using bidirectional LSTMs to improve the performance of *stack-LSTM* model which learn representations of the parser state. In Kiperwasser and Goldberg (2016) the bidirectional LSTMs recurrent output vector for each word is concatenated with any possible heads recurrent vector, and the result is used as input to a

[1] http://www.evalita.it

multi-layer perceptron (MLP) network that scores each resulting edge. Cheng et al. (2016) propose a bidirectional attention model which uses two additional unidirectional RNN, called left-right and right-left query component. Based on Kiperwasser and Goldberg (2016) and Cheng et al. (2016) model, in Dozat and Manning (2017) a biaffine attention mechanism is used, instead of traditional MLP-based attention. The model proposed in Nguyen et al. (2017) train a neural network model that learn jointly POS tagging and graph-based dependency parsing. The model uses a bidirectional LSTM to learn POS tagging and the Kiperwasser and Goldberg (2016) approach for dependency parsing. Shi et al. (2017a,b) described a parser that combines three parsing paradigms using a dynamic programming approach.

Parser Ref.-Abbreviation	Method	Parsing
(Chen and Manning, 2014) - CM14	Tb: a-s	Greedy
(Ballesteros et al., 2015) - BA15	Tb: a-s	Be-se
(Kiperwasser and Goldberg, 2016)- KG16:T	Tb: a-h	Greedy
(Kiperwasser and Goldberg, 2016)- KG16:G	Gb: a-f	Eisner
(Andor et al., 2016) - AN16	Tb: a-s	Beam-S
(Cheng et al., 2016) - CH16	Gb: a-f	cle
(Dozat and Manning, 2017) - DM17	Gb: a-f	cle
(Shi et al., 2017a,b)- SH17	Tb: a-h./ -eager	Greedy
	Gb: a-f	Eisner
(Nguyen et al., 2017) - NG17	Gb: a-f	Eisner

Table 1: All the neural parsers considered in this study with their fundamental features as well as their abbreviations used throughout the paper. In this table "Tb/Gb" means "Transition/Graph-based", "Beam-S" means "Beam-search" and "a-s/h/f" means "arc-standard/hybrid/factored".

We trained, validated and tested the nine considered parsers, as well as all the proposed extensions, by considering three different setups:

- **setup0**: only the UD Italian 2.1 dataset;
- **setup1**: only the UD Italian PoSTWITA 2.2 dataset;
- **setup2**: UD Italian 2.1 dataset joined with the UD Italian PoSTWITA 2.2 dataset (train and validation sets) keeping the test set of PoSTWITA 2.2;

After the influential paper from Reimers and Gurevych (2017) it is clear to the community that reporting a single score for each DNN training session could be heavily affected by the system initialisation point and we should instead report the mean and standard deviation of various runs with the same setting in order to get a more accurate picture of the real systems performances and make more reliable comparisons between them.

Table 2 shows the parsers performances on the test set for the three setups described above executing the training/validation/test cycle for 5 times. In any setup the DM17 parser exhibits the best performances, notably very high for general Italian. As we can expect, the performances on setup1 were much lower than that for setup0 due to the intrinsic difficulties of parsing tweets and to the scarcity of annotated tweets for training. Joining the two datasets in the setup2 allowed to get a relevant gain in parsing tweets even if we added out-of-domain data. For these reasons, for all the following experiments, we abandoned the setup1 because it seemed more relevant to use the joined data (setup2) and compare them to setup0.

3 An Ensemble of Neural Parsers

The DEPENDABLE tool in Choi et al. (2015) reports ensemble upper bound performance assuming that, given the parsers outputs, the best tree can be identified by an oracle "MACRO" (MA), or that the best arc can be identified by another oracle "MICRO" (mi). Table 3 shows that, by applying these oracles, we have plenty of space for improving the performances by building some kind of ensemble system able to cleverly choose the correct information from the different parsers outputs and combine them improving the final solution. This observation motivates our proposal.

To combine the parser outputs we used the following ensemble schemas:

- **Voting**: Each parser contributes by assigning a vote on every dependency edge as described in Zeman and Žabokrtský (2005). With the majority approach the dependency tree could be ill-formed, in this case using the switching approach the tree is replaced with the output of the first parser.
- **Reparsing**: As described in Sagae and Lavie (2006) together with Hall et al. (2007) a MST algorithm is used to reparse a graph where

	setup0			
	Valid. Ita		**Test Ita**	
	UAS	LAS	UAS	LAS
CM14	88.20/0.18	85.46/0.14	89.33/0.17	86.85/0.22
BA15	91.15/0.11	88.55/0.23	91.57/0.38	89.15/0.33
KG16:T	91.17/0.29	88.42/0.24	91.21/0.33	88.72/0.24
KG16:G	91.85/0.27	89.23/0.31	92.04/0.18	89.65/0.10
AN16	85.52/0.34	77.67/0.30	87.70/0.31	79.48/0.24
CH16	92.42/0.00	89.60/0.00	92.82/0.00	90.26/0.00
DM17	**93.37**/0.27	**91.37**/0.24	**93.72**/0.14	**91.84**/0.18
SH17	89.67/0.24	85.05/0.24	89.89/0.29	84.55/0.30
NG17	90.37/0.12	87.19/0.21	90.67/0.15	87.58/0.11
	setup1			
	Valid. PoSTW		**Test PoSTW**	
	UAS	LAS	UAS	LAS
CM14	81.03/0.17	75.24/0.30	81.50/0.28	76.07/0.17
BA15	83.44/0.20	77.70/0.25	84.06/0.38	78.64/0.44
KG16:T	77.38/0.14	68.81/0.25	77.41/0.43	69.13/0.43
KG16:G	78.81/0.23	70.14/0.33	78.78/0.44	70.52/0.51
AN16	77.74/0.25	66.63/0.16	77.78/0.33	67.21/0.30
CH16	84.78/0.00	78.51/0.00	86.12/0.00	79.89/0.00
DM17	**85.01**/0.16	**78.80**/0.09	**86.26**/0.16	**80.40**/0.19
SH17	80.52/0.18	73.71/0.14	81.11/0.29	74.53/0.26
NG17	82.02/0.11	75.20/0.24	82.74/0.39	76.22/0.41
	setup2			
	Valid. Ita+PoSTW		**Test PoSTW**	
	UAS	LAS	UAS	LAS
CM14	85.52/0.13	81.51/0.05	82.62/0.24	77.45/0.23
BA15	87.85/0.13	83.80/0.12	85.15/0.29	80.12/0.27
KG16:T	83.89/0.23	77.77/0.26	80.47/0.36	72.92/0.46
KG16:G	84.70/0.14	78.41/0.14	81.41/0.37	73.49/0.19
AN16	82.95/0.33	73.46/0.37	79.81/0.27	69.19/0.19
CH16	89.16/0.00	84.56/0.00	86.85/0.00	80.93/0.00
DM17	**89.72**/0.10	**85.85**/0.13	**87.22**/0.24	**81.65**/0.21
SH17	85.85/0.36	80.00/0.39	83.12/0.50	76.38/0.38
NG17	86.81/0.04	82.13/0.09	84.09/0.07	78.02/0.11

Table 2: Mean/standard deviation of UAS/LAS for each parser and for the different setups by repeating the experiments 5 times. All the results are statistically significant ($p < 0.05$) and the best values are showed in boldface.

	Validation		**Test**	
	UAS	LAS	UAS	LAS
	setup0			
mi	98.30%	97.82%	98.08%	97.72%
MA	96.62%	95.10%	96.31%	94.82%
	setup2			
mi	97.08%	96.02%	96.32%	94.73%
MA	94.62%	91.29%	93.27%	88.50%

Table 3: Results obtained by building an ensemble system based on the oracles mi e MA and considering all parsers.

each word in the sentence is a node. The MSTs algorithms used are Chu-Liu/Edmons (cle) and Eisner as reported in McDonald et al. (2005). Three weighting strategies for Chu-Liu/Edmons are used: equally weighted (w2); weighted according to the total labeled accuracy on the validation set (w3); weighted according to labeled accuracy per coarse grained PoS tag on the validation set (w4).

- **Distilling**: In Kuncoro et al. (2016) the authors train a distillation parser using a loss objective with a cost that incorporates ensemble uncertainty estimates for each possible attachment.

4 Results

Tables 4, 7 and 9 show the performances of the ensembles built on the best results on validation set obtained in the 5 training/test cycles considering both setup0 and setup2. Table 6 reports the number of malformed trees for the majority strategy.

Table 5 and 8 report the number of cases when the ensemble combination output differs from the baseline, including both labeled (L) and unlabeled (U) outputs. On the average the percentage of different unlabeled output varies from 2% to 15% with respect to baseline. For the best result (DM17+ALL) the difference on setup0 and setup2 is about 4%.

The results of the voting approach reported in Table 4 shows that the majority strategy is slightly better than the switching strategy, although it must be taken into account that there might be ill-formed dependency trees for the former strategy. The percentage of ill-formed trees on valid./test set vary from a minimum of 2% to a maximum of 8%. For this reasons the majority strategy should be used when it is followed by a manual correction phase. The switching strategy performs well if the first parser of voters is one of the best parsers, in fact the combinations AN16+ALL and AN16+CM14+SH17 have worst performance than the counterparts which using the best parser (DM17) as the first voter. Overall, the highest performance is achieved using all parsers together with DM17 as the first voter. For setup0 the increases are +0.19% in UAS e +0.38% in LAS, while in setup2 are +0.92% in UAS e +2.47% in LAS with respect to the best single parser (again DM17).

The results of the reparsing approach reported in Table 7 shows that the Chu-Liu/Edmonds algorithm is slightly better than the Eisner algorithm. In this case, the choice of which strategy

setup0				
	Validation		**Test**	
Voters/Strategy	UAS	LAS	UAS	LAS
DM17+CH16+BA15/maj.	94.20%	92.27%	93.77%	92.13%
DM17+CH16+BA15/swi.	94.11%	92.16%	93.79%	92.14%
AN16+CM14+SH17/maj.	90.43%	87.96%	91.03%	88.47%
AN16+CM14+SH17/swi.	89.44%	86.77%	90.17%	87.43%
DM17+CM14+SH17/maj.	93.84%	92.03%	93.82%	92.27%
DM17+CM14+SH17/swi.	93.76%	91.94%	93.82%	92.25%
AN16+ALL/maj.	94.37%	92.65%	93.83%	92.27%
AN16+ALL/swi.	93.99%	92.15%	93.43%	91.73%
DM17+ALL/maj.	**94.42%**	**92.67%**	**93.94%**	**92.41%**
DM17+ALL/swi.	94.38%	92.60%	93.91%	92.37%
DM17 (baseline)	93.74%	91.66%	93.75%	92.03%
setup2				
	Validation		**Test**	
Voters/Strategy	UAS	LAS	UAS	LAS
DM17+CH16+BA15/maj.	90.57%	87.16%	88.21%	83.64%
DM17+CH16+BA15/swi.	90.51%	87.10%	88.13%	83.51%
AN16+CM14+SH17/maj.	86.90%	83.60%	84.09%	79.78%
AN16+CM14+SH17/swi.	86.01%	82.50%	82.58%	77.94%
DM17+CM14+SH17/maj.	90.35%	87.21%	88.07%	83.64%
DM17+CM14+SH17/swi.	90.27%	87.11%	87.99%	83.52%
AN16+ALL/maj.	90.30%	87.26%	88.36%	84.13%
AN16+ALL/swi.	89.70%	86.45%	87.46%	83.06%
DM17+ALL/maj.	90.64%	87.60%	**88.51%**	**84.42%**
DM17+ALL/swi.	**90.65%**	**87.62%**	88.50%	84.20%
DM17 (baseline)	89.82%	85.96%	87.59%	81.95%

Table 4: Results of ensembles using switching and majority approaches on the best models in setup0 and setup2. The baseline is defined by the best results of Dozat and Manning (2017).

setup0				
	Validation /11.908		**Test** /10.417	
Voters/Strategy	U	L	U	L
DM17+CH16+BA15/maj.	208	61	188	46
DM17+CH16+BA15/swi.	192	52	175	39
AN16+CM14+SH17/maj.	1.006	424	783	336
AN16+CM14+SH17/swi.	1.130	489	870	371
DM17+CM14+SH17/maj.	170	37	139	15
DM17+CM14+SH17/swi.	157	33	129	13
AN16+ALL/maj.	382	126	328	105
AN16+ALL/swi.	460	164	386	133
DM17+ALL/maj.	356	117	282	81
DM17+ALL/swi.	312	97	255	72
setup2				
	Validation /24.243		**Test** /12.668	
Voters/Strategy	U	L	U	L
DM17+CH16+BA15/maj.	597	219	470	213
DM17+CH16+BA15/swi.	521	185	394	172
AN16+CM14+SH17/maj.	2.757	1.329	1.805	941
AN16+CM14+SH17/swi.	2.976	1.429	1.986	1.033
DM17+CM14+SH17/maj.	490	140	337	93
DM17+CM14+SH17/swi.	453	121	300	73
AN16+ALL/maj.	1.377	624	897	440
AN16+ALL/swi.	1.610	741	1.063	534
DM17+ALL/maj.	1.156	502	784	378
DM17+ALL/swi.	920	374	614	280

Table 5: Numbers of cases when there is a different output between the ensemble systems, using switching and majority, and the baseline Dozat and Manning (2017).

	setup0		setup2	
Voters	**Valid.** /564	**Test** /482	**Valid.** /1235	**Test** /674
DM17+CH16+BA15	9	7	31	31
AN16+CM14+SH17	45	25	88	77
DM17+CM14+SH17	6	6	19	23
AN16+ALL	18	17	73	63
DM17+ALL	17	11	75	57

Table 6: Number of malformed trees obtained by using the majority strategy for both setups.

to use must take into account if we want to allow non-projectivity or not. The percentage of non-projective dependency trees on valid./test set for Chu-Liu/Edmonds vary from a minimum of 7% to a maximum of 12% compared with the average for the Italian corpora of 4%. Overall, the highest performances are achieved using Chu-Liu/Edmonds algorithm. For setup0 the increases are +0.25% in UAS and +0.45% in LAS, while in setup2 are +0.77% in UAS and +2.30% in LAS with respect to the best single parser (DM17).

The results of the distilling strategy reported in Table 9, unlike the previous proposals, show worse outcomes, which score below the baseline.

5 Discussion and Conclusions

We have studied the performances of some neural dependency parsers on generic and social media domain. Using the predictions of each single parser we combined the best outcomes to improve the performance in various ways. The ensemble models are more efficient on corpora built using in-domain data (social media), giving an improvement of $\sim 1\%$ in UAS and $\sim 2.5\%$ in LAS.

Thanks to the number of parser models adopted in the experiments it has been possible to verify that the performances of the ensemble models increase as the number of parsers grows.

The improvement of LAS is, in most cases, at least twice the value of UAS. This could mean that ensemble models catch with better precision the type of dependency relations rather than head-dependent relations.

All the proposed ensemble strategies, except for distilling, perform more or less in the same way, therefore the choice of which strategy to use is due, in part, to the properties that we want to obtain on the combined dependency tree.

Our work is inspired by the work of Mazzei

setup0				
	Validation		**Test**	
Voters/Strategy	UAS	LAS	UAS	LAS
DM17+CH16+BA15/cle-w2	93.82%	91.85%	93.54%	91.83%
DM17+CH16+BA15/cle-w3	93.89%	91.82%	93.78%	92.06%
DM17+CH16+BA15/cle-w4	94.20%	92.28%	93.72%	92.04%
DM17+CH16+BA15/eisner	94.05%	92.05%	93.46%	91.78%
ALL/cle-w2	**94.31**%	92.53%	93.85%	92.23%
ALL/cle-w3	94.16%	92.41%	**94.00**%	**92.48**%
ALL/cle-w4	94.29%	**92.58**%	93.95%	92.38%
ALL/eisner	**94.31**%	92.53%	93.95%	92.35%
DM17 (baseline)	93.74%	91.66%	93.75%	92.03%
setup2				
	Validation		**Test**	
Voters/Strategy	UAS	LAS	UAS	LAS
DM17+CH16+BA15/cle-w2	90.33%	86.95%	87.69%	83.31%
DM17+CH16+BA15/cle-w3	89.82%	85.96%	87.59%	81.95%
DM17+CH16+BA15/cle-w4	90.41%	86.99%	87.94%	83.32%
DM17+CH16+BA15/eisner	90.50%	87.05%	88.04%	83.51%
ALL/cle-w2	**90.52**%	**87.53**%	**88.36**%	**84.25**%
ALL/cle-w3	89.90%	86.75%	87.79%	83.54%
ALL/cle-w4	90.42%	87.46%	88.19%	84.11%
ALL/eisner	90.45%	87.41%	88.31%	84.08%
DM17 (baseline)	89.82%	85.96%	87.59%	81.95%

Table 7: Results of ensembles using reparsing approaches on the best models in setup0 and setup2. The baseline is again defined by the best results of DM17.

setup0				
	Validation /11.908		**Test** /10.417	
Voters/Strategy	UAS	LAS	UAS	LAS
DM17+CH16+BA15/cle-w2	360	129	307	90
DM17+CH16+BA15/cle-w3	96	0	89	1
DM17+CH16+BA15/cle-w4	267	76	247	52
DM17+CH16+BA15/eisner	375	130	327	103
ALL/cle-w2	400	131	333	103
ALL/cle-w3	351	108	299	79
ALL/cle-w4	383	126	307	87
ALL/eisner	411	133	333	106
setup2				
	Validation /24.243		**Test** /12.668	
Voters/Strategy	UAS	LAS	UAS	LAS
DM17+CH16+BA15/cle-w2	1.056	496	800	424
DM17+CH16+BA15/cle-w3	0	0	0	0
DM17+CH16+BA15/cle-w4	603	264	491	236
DM17+CH16+BA15/eisner	1.047	443	789	376
ALL/cle-w2	1.347	599	882	417
ALL/cle-w3	1.261	537	804	363
ALL/cle-w4	1.274	576	822	389
ALL/eisner	1.367	607	916	436

Table 8: Numbers of cases when there is a different output between the ensemble systems, using reparsing approaches, and the baseline Dozat and Manning (2017).

(2015). Different from his work, we use larger set of state-of-the-art parsers, all based on neural networks, in order to gain more diversity among

Setup	UAS	LAS
setup0	92.50% (–1.25%)	89.93% (–2.10%)
setup2	86.73% (–0.86%)	81.39% (–0.56%)

Table 9: Results of distilling approach on the best models in setup0 and setup2. In brackets are reported the differences between the distilled models and the best results of DM17, as baseline.

the models used in the ensembles; furthermore we have experimented the distilling strategy and eisner reparsing algorithm. Moreover, we built ensembles on larger datasets using both generic and social media texts.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

Anita Alicante, Cristina Bosco, Anna Corazza, and Alberto Lavelli. 2015. Evaluating italian parsing across syntactic formalisms and annotation schemes. In Roberto Basili, Cristina Bosco, Rodolfo Delmonte, Alessandro Moschitti, and Maria Simi, editors, *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*, Springer International Publishing, Cham, pages 135–159.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ACL, Berlin, Germany, pages 2442–2452.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. ACL, Lisbon, Portugal, pages 349–359.

Cristina Bosco, Felice DellOrletta, Simonetta Montemagni, Manuela Sanguinetti, and Maria Simi. 2014. The evalita 2014 dependency parsing task. In *Proceedings of the Fourth Inter-*

national Workshop EVALITA 2014. Pisa, Italy, pages 1–8.

Cristina Bosco and Alessandro Mazzei. 2011. The evalita 2011 parsing task. In *Working Notes of EVALITA 2011*, CELCT, Povo, Trento.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, Doha, Qatar, pages 740–750.

Hao Cheng, Hao Fang, Xiaodong He, Jianfeng Gao, and Li Deng. 2016. Bi-directional attention with agreement for dependency parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. ACL, Austin, Texas, pages 2204–2214.

Jinho D. Choi, Joel Tetreault, and Amanda Stent. 2015. It depends: Dependency parser comparison using a web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. ACL, Beijing, China, pages 387–396.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of the 2017 International Conference on Learning Representations*.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. ACL, Beijing, China, pages 334–343.

Johan Hall, Jens Nilsson, Joakim Nivre, Gülsen Eryigit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single malt or blended? a study in multilingual parser optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*. ACL, Prague, Czech Republic, pages 933–939.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics* 4:313–327.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, and Noah A. Smith. 2016. Distilling an ensemble of greedy dependency parsers into one mst parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. ACL, Austin, Texas, pages 1744–1753.

Alberto Lavelli. 2013. An ensemble model for the evalita 2011 dependency parsing task. In Bernardo Magnini, Francesco Cutugno, Mauro Falcone, and Emanuele Pianta, editors, *Evaluation of Natural Language and Speech Tools for Italian*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 30–36.

Alberto Lavelli. 2014. Comparing state-of-the-art dependency parsers for the evalita 2014 dependency parsing task. In *Proceedings of the Fourth International Workshop EVALITA 2014*. Pisa, Italy, pages 15–20.

Alberto Lavelli. 2016. Comparing state-of-the-art dependency parsers on the italian stanford dependency treebank. In *Proceedings of the Third Italian Conference on Computational Linguistics (CLiC-it 2016)*. Napoli, Italy, pages 173–178.

Alessandro Mazzei. 2015. Simple voting algorithms for italian parsing. In Roberto Basili, Cristina Bosco, Rodolfo Delmonte, Alessandro Moschitti, and Maria Simi, editors, *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*, Springer International Publishing, Cham, pages 161–171.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. ACL, Vancouver, British Columbia, Canada, pages 523–530.

Dat Quoc Nguyen, Mark Dras, and Mark Johnson. 2017. A novel neural network model for joint pos tagging and graph-based dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. ACL, Vancouver, Canada, pages 134–142.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. ACL, Copenhagen, Denmark, pages 338–348.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. ACL, Stroudsburg, PA, USA, NAACL-Short '06, pages 129–132.

Tianze Shi, Liang Huang, and Lillian Lee. 2017a. Fast(er) exact decoding and global training for transition-based dependency parsing via a minimal feature set. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. ACL, Copenhagen, Denmark, pages 12–23.

Tianze Shi, Felix G. Wu, Xilun Chen, and Yao Cheng. 2017b. Combining global models for parsing universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. ACL, Vancouver, Canada, pages 31–39.

Daniel Zeman and Zdeněk Žabokrtský. 2005. Improving parsing accuracy by combining diverse dependency parsers. In *Proceedings of the Ninth International Workshop on Parsing Technology*. ACL, Vancouver, British Columbia, pages 171–178.