Towards Making a Dependency Parser See

Michalina Strzyz   **David Vilares**   **Carlos Gómez-Rodríguez**
Universidade da Coruña, CITIC
FASTPARSE Lab, LyS Research Group, Departamento de Computación
Campus de Elviña, s/n, 15071 A Coruña, Spain
{michalina.strzyz,david.vilares,carlos.gomez}@udc.es

Abstract

We explore whether it is possible to leverage eye-tracking data in an RNN dependency parser (for English) when such information is only available during training – i.e. no aggregated or token-level gaze features are used at inference time. To do so, we train a multi-task learning model that parses sentences as sequence labeling and leverages gaze features as auxiliary tasks. Our method also learns to train from disjoint datasets, i.e. it can be used to test whether already collected gaze features are useful to improve the performance on new *non-gazed* annotated treebanks. Accuracy gains are modest but positive, showing the feasibility of the approach. It can serve as a first step towards architectures that can better leverage eye-tracking data or other complementary information available only for training sentences, possibly leading to improvements in syntactic parsing.

1 Introduction

Eye trackers and gaze features collected from them have been recently applied to natural language processing (NLP) tasks, such as part-of-speech tagging (Duffy et al., 1988; Nilsson and Nivre, 2009; Barrett and Søgaard, 2015a), named-entity recognition (Tokunaga et al., 2017) or readability (González-Garduño and Søgaard, 2018). Eye-movement data has been also used for parsing. For example, Barrett and Søgaard (2015b) rank discriminative features to predict syntactic categories (e.g. subject vs. object) and use them to improve a transition-based parser, trained on a structured perceptron with discrete features (Collins, 2002; Zhang and Nivre, 2011). However, the experiments were carried out on a parallel toy treebank and the performance was relatively low. Lopopolo et al. (2019) follow the inverse path, and use dependency parsing features to predict eye-

regression during training, i.e. cases where the reader goes back to a word of the sentence.

In this context, how to retrieve and leverage eye-tracking data has become an active area of research in different NLP fields. Previous studies (Barrett and Søgaard, 2015b; Barrett et al., 2016) suggest that real-time eye-tracking data can be collected at inference time, so that *token-level* gaze features are used during training but also at test time. However, even if in the near future every user has an eye tracker on top of their screen – a scenario which is far from guaranteed, and raises privacy concerns (Liebling and Preibusch, 2014) – many running NLP applications that process data from various Internet sources will not expect to have any human being reading massive amounts of data. Other studies (Barrett and Søgaard, 2015a; Hollenstein and Zhang, 2019) instead derive gaze features from the training set: in particular, they collect *type-level* gaze features from the vocabulary in the training set; which are then aggregated to create a lookup table and used as a sort of precomputed gaze input when a given word in the test set matches an entry, otherwise, a token has an unknown gaze feature value. In this manner, the influence of gaze features on unseen data depends on the vocabulary encountered during training. For instance, Hollenstein et al. (2019) report that unknown tokens make up 41.09% on new data for their research on named-entity recognition (NER) with eye-tracking data. To our knowledge, this kind of approaches have not been applied to syntactic parsing.

More in the line of our work, Barrett et al. (2018) is one of the few approaches that does not rely on the assumption of using gaze features as input. Instead, the human data is used as an inductive bias to guide the attention weights in a recurrent neural network for sequence classification (used for tasks such as binary sentiment anal-

1500

ysis a the sentence level). Moreover, it has been shown that both constituency (Vilares et al., 2019) and dependency (Strzyz et al., 2019a) parsing can benefit from multi-task learning. Also related to our work with human data, González-Garduño and Søgaard (2017) show that gaze features learned in a multi-task (MTL) setup can lead to improvements in readability assessment.

Contribution In this work we leverage gaze learning for dependency parsing assuming that gaze features are likely to be only available during training – and no gaze features are given, in any form, at inference time. Our approach learns eye-movement features as auxiliary tasks in a multitask framework where both parsing and gaze prediction are addressed as sequence labeling. To test its effect on syntactic analysis, we experiment on parallel and disjoint datasets with dependency parsing and gaze annotations. The source code can be found at `https://github.com/mstrise/dep2label-eye-tracking-data`.

2 Methods and materials

This section describes in detail the data and the models used in our work.

2.1 Data

We will use both *parallel* data (containing dependency and gaze annotations) and *disjoint* data (where one dataset contains only parsing annotations and the other one just gaze data).

***Parallel* data** The Dundee corpus (Kennedy et al., 2003) contains recordings of measurements for eye movements of ten English-speaking participants during reading 20 newspaper articles from *The Independent* making in total 2368 unique sentences. This dataset was used as the starting point to create a corpus that fits the purpose of dependency parsing, the Dundee treebank (Barrett et al., 2015). This treebank augments sentences from the Dundee corpus with syntactic annotations according to the Universal Dependencies (UD) guidelines (Nivre et al., 2016). We will use this treebank as the parallel data for our models. We split the data $(80 - 10 - 10)$ into training, development and test. Sentences were randomly shuffled with assurance that the same sentence coming from the

10 participants is included only in one of the sets.[1]

***Disjoint* data** Most of dependency treebanks do not contain gaze annotations. With this setup, in addition to the assumption that eye-movement data is available during training, we aim to show whether we improve the performance of a dependency parser with gaze-annotated data coming from a different corpus. We will use two datasets: one with dependency annotations and another one labeled with eye-movement data. For the former we use the Penn Treebank (PTB) (Marcus et al., 1993), which contains sentences from *The Wall Street Journal* annotated with phrase-structure trees. We convert it into Stanford Dependencies (de Marneffe et al., 2006) and apply the standard splits for dependency parsing: sections $2 - 21$ for training, 22 for development set and 23 for testing, whereas PoS tags are predicted by the Stanford tagger (Toutanova et al., 2003). For gaze-annotated data, we will employ the eye-movement annotations from the Dundee Treebank but using a different split $(90 - 10 - 0)$.[2]

2.2 Gaze-averaged sequence labeling parsing

During training we opt to emulate recent cognitive studies that suggest that human sentence understanding recruits the same brain regions for lexical and syntactic processing, consistent with a sequence-tagging-like process (Fedorenko et al., 2018), which is also a natural and intuitive way to learn to predict eye-movement information. To do so, we rely on standard bidirectional long short-term memory networks (BILSTMs) (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997).

We denote $\text{LSTM}_\theta(\mathbf{w})$ as the abstraction of a long short-term memory network, that processes an input sentence $w = [w_1, ..., w_n]$ to generate an output of hidden representations $\mathbf{h} = [\mathbf{h}_1, ..., \mathbf{h}_n]$. Thus, a BILSTM can be seen as $\text{BILSTM}_\theta(\mathbf{w}) = \text{LSTM}_\theta^l(\mathbf{w_{i:m}}) \cdot \text{LSTM}_\theta^r(\mathbf{w_{m:n}}) = \mathbf{h}^l \cdot \mathbf{h}^r$.[3] In this work we will stack two layers of BILSTMs before decoding the output. BILSTMs are commonly used for sequence labeling tasks (Reimers

---

and Gurevych, 2017), where a word-level prediction can be generated for each token, by adding a feed forward network to predict an output label at each time step, using a $softmax(\mathbf{W} \times \mathbf{h}_i + \mathbf{b})$. In this paper we propose to learn to leverage eye-movement data as a sequence labeling task. To jointly learn dependency parsing in a common framework, we also cast parsing as sequence labeling and learn both tasks in a multitask learning (MTL) setup (Caruana, 1997). The main components are described in the following paragraphs.

Dependency parsing as sequence labeling We proceed similarly to Strzyz et al. (2019b). Given a linearization function $F_w : T_{|w|} \to L^w$, for each word w_i, Strzyz et al.'s approach generates a label $l_i \in (o_i, p_i, d_i)$ that encodes the binary relationship between w_i and its head, where: d_i encodes the dependency relation, and the sub-pair (o_i, p_i) the index of such head term, with $o_i \in \mathbb{N}$ and $p_i \in P$ (a part-of-speech set). If o_i is positive, then the head of w_i is the o_i-th token to the right that has the part-of-speech tag p_i. If o_i is negative, then the head of w_i is the $|o_i|$-th token to the left whose PoS tag is p_i.

This parser obtains similar results to competitive transition- and graph-based parsers such as BIST (Kiperwasser and Goldberg, 2016) and can be taken as a strong baseline to test the effect of eye-movement data for dependency parsing.

Gaze information In previous studies the choice of number of gaze features used in the experiments has varied, seemingly, depending on the NLP task of interest. For instance, Barrett et al. (2016) distinguish 31 features (where 22 are gaze features) for part-of-speech tagging while Hollenstein and Zhang (2019) use 17 features in the NER task. In another piece of work, 5 gaze features are used for relation classification and sentiment analysis (Hollenstein et al., 2019). Finally, Singh et al. (2016) use gaze features in order to automatically predict reading times for a new text. However, the model predicts only 4 features that are then used as features for readability assessment.

We have chosen 12 gaze features and based on the previous work (Barrett et al., 2016; Hollenstein and Zhang, 2019) we have divided them into 4 groups. In particular, we explore the informativeness of the *basic gaze features*: total fixation duration on a word w (`total fix dur`), mean fixation duration on w (`mean fix dur`, num-

ber fixations on w (n `fix`) and fixation probability on w (`fix prob`). As *early gaze features* we consider: first fixation duration on w (`first fix dur`) and first pass duration (`first pass dur`), while as *late gaze features*: number of re-fixations on w (n `re-fix`) and reread probability of w (`reread prob`). We also take account of the neighboring words and treat them as *context features*: fixation probability on the previous and next word ($w - 1$ and $w + 1$ `fix prob`) as well as fixation duration on the previous and next word ($w - 1$ and $w + 1$ `fix dur`). Figure 1 depicts an exemplary (linearized) dependency tree with some of the gaze features used in the experiments. Following previous work (Hollenstein and Zhang, 2019), we discretize gaze features.[4]

Multitask learning We previously argued that gaze-annotated data is unlikely to be available at inference time. However, previous related work for dependency parsing usually assumes that such information will be collected during testing and fed as input features to the model (Barrett and Søgaard, 2015b; Hollenstein and Zhang, 2019). What we propose instead is to leverage such information during training. To do so, we rely on MTL and auxiliary tasks. Our work focuses on exploiting the utility of gaze information using just a standard BILSTM, directly building on top of previous work of dependency parsing as sequence labeling (Strzyz et al., 2019b), and ignoring extra tools such as attention. In this line, a future possible solution could be to apply the approach by Barrett et al. (2018) to structured prediction and word-level classification. In their work they used human data as an inductive bias to update the attention weights of the network.

In our setup, dependency parsing as sequence labeling is addressed through two main tasks: one to predict the index of the head term, i.e. the sub-index (o_i, p_i), and another one to predict the dependency relation (d_i). Eye-movement discrete labels are learned as auxiliary tasks. We use a hard-sharing architecture where the BILSTMs are fully shared and followed by an independent feed-forward layer (followed by a softmax) to predict the corresponding label for each of the tasks. The main idea is that the eye-movement signal(s) will

---

[4]`Total fix dur`, `mean fix dur`, `first fix dur`, `first pass dur`, $w - 1$ and $w + 1$ `fix dur` are used as a discrete variable whose values are represented as percentile intervals with a bin size of 20, while for the other features we use their raw values.
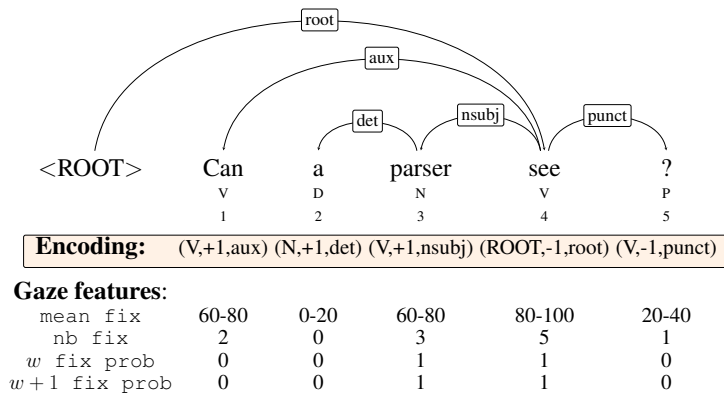
Figure 1: A dependency tree with its encoding and some of the corresponding gaze features.

be back-propagated to update the weights of the shared BILSTM, building a model that latently encodes such information and helps dependency parsing.

For the parallel data setup, the model is trained in a standard way, and the cross-entropy loss is computed as $\mathcal{L} = \mathcal{L}_{(o,p)} + \mathcal{L}_d + \sum_a \beta_a * \mathcal{L}_a$. For the disjoint setup, during the training process for each batch we randomly pick all its samples from one of the treebanks (samples that have not been yet taken, either from the dependency parsing corpus or the gaze-annotated one) and run the network to make the predictions. Then, we back-propagate the loss for the outputs associated to that treebank and ignore the rest, i.e. $\mathcal{L} = \tau(\mathcal{L}_{(o,p)} + \mathcal{L}_d) + (1 - \tau) \sum \beta_{aux} \mathcal{L}_{aux}$, where τ is a binary flag set to 1 when the batch contains dependency parsing samples, and to 0 otherwise.

For both setups we will test: (i) predicting *one* of the gaze features as an auxiliary task (ii) predicting multiple gaze labels as multiple auxiliary tasks.

3 Experiments

We conduct an explanatory study for the general viability of our method. We will compare results on the development and test set to verify whether the improvements are consistent. In particular, we carry out two experiments:

- Experiment 1 (on parallel data): Tests how eye-tracking data influences the performance when both gaze features and syntactic dependencies are extracted from the same data (in this case the Dundee treebank). The baseline is a model where no gaze features were used.

- Experiment 2 (on disjoint data):

As previously said, we wish to test whether we can leverage existing gaze-annotated datasets to improve the performance in other non-gazed annotated dependency treebanks. Also, note that the Dundee treebank is relatively small ($\sim$2k sentences) and results from the previous Experiment can be not as representative as those obtained in larger corpora. In this setup, we train models using gaze information from the Dundee treebank and dependency relations from PTB, evaluating on the latter.[5] The baseline is obtained by training a model only on the dependency relations retrieved from PTB.

Metrics We evaluate the parsers with Unlabeled and Labeled Attachment Score (UAS and LAS) excluding punctuation (following the standard methodology for experiments on the PTB).

3.1 Results

Table 1 shows the results for Experiment 1 (on the Dundee treebank, i.e. on parallel data), suggesting that the degree of informativeness of the gaze features that a parser can leverage differ among each other. In particular, we observe that the improvements across different models and the development and test sets are unstable. The evaluation on the dev set shows that gaze features from *early* and *context* group used as multiple auxiliary tasks (`early` and `context feats aux`) modestly improve the model in comparison with the baseline. With respect to the results on the test set, the most informative features are: `mean fix dur`,

---

[5]Note that the set of fine-tuned embeddings can be slightly different from that of Experiment 1, since words not occurring in Experiment 1 might be present in Experiment 2 due to the use of a different corpus.

1503

Gaze features		dev set		test set	
		UAS	LAS	UAS	LAS
	baseline	85.36	79.40	84.37	78.24
Basic	`total fix dur`	85.34	79.35	84.06	77.44
	`mean fix dur`	85.21	79.38	84.59	**78.70**
	n `fix`	85.32	79.29	83.71	77.57
	`fix prob`	85.32	79.57	84.33	77.91
	`basic feats aux`	85.36	79.57	83.86	77.75
Early	`first fix dur`	85.30	79.46	**84.64**	78.57
	`first pass dur`	85.50	79.49	84.55	78.39
	`early feats aux`	**85.61**	79.57	84.37	78.11
Late	n `re-fix`	85.52	79.25	83.86	77.91
	`reread prob`	85.34	79.57	83.86	77.37
	`late feats aux`	85.54	79.64	84.10	77.73
Context	$w-1$ `fix prob`	85.17	79.47	84.26	77.93
	$w+1$ `fix prob`	85.36	79.07	84.24	78.06
	$w-1$ `fix dur`	85.43	79.68	84.50	77.95
	$w+1$ `fix dur`	85.39	79.53	**84.64**	78.30
	`context feats aux`	**85.61**	**79.72**	84.33	78.24

Table 1: Impact of various gaze features as auxiliary task(s) on the score (UAS/LAS) of dependency parsing as the main task evaluated on Dundee treebank (parallel setup).

Gaze features		dev set		test set	
		UAS	LAS	UAS	LAS
	baseline	93.98	91.67	93.86	91.80
Basic	`total fix dur`	93.94	91.60	93.99	91.92
	`mean fix dur`	**94.12**	**91.84**	93.95	91.82
	n `fix`	93.97	91.70	93.91	91.87
	`fix prob`	93.98	91.71	93.99	91.93
	`basic feats aux`	94.00	91.69	93.84	91.81
Early	`first fix dur`	94.07	91.81	93.87	91.80
	`first pass dur`	93.93	91.58	93.79	91.70
	`early feats aux`	94.04	91.78	93.96	91.88
Late	n `re-fix`	94.01	91.69	93.87	91.79
	`reread prob`	94.03	91.74	93.89	91.89
	`late feats aux`	93.98	91.58	93.92	91.90
Context	$w-1$ `fix prob`	94.02	91.65	93.95	91.93
	$w+1$ `fix prob`	93.88	91.61	93.89	91.82
	$w-1$ `fix dur`	94.06	91.65	93.86	91.83
	$w+1$ `fix dur`	93.91	91.69	93.89	91.84
	`context feats aux`	93.93	91.63	**94.01**	**91.98**

Table 2: Results for dependency parsing evaluated on PTB treebank with gaze features as auxiliary task(s) learned from the disjoint dataset: Dundee treebank.

which improves the LAS by +0.46, and `first fix dur` by +0.33.

Table 2 shows the results for Experiment 2 (using Dundee gaze data and PTB dependencies). Under this setup, the gains decrease in comparison with the results on the parallel setup. This could be partially caused by not using parallel data. The improvements seem to be more consistent between the development and test sets. This could be related to the fact that we use a larger (more representative) treebank. When looking at the dev set scores, the most discriminative gaze feature is `mean fix dur` that increases LAS by +0.17 and `first fix dur` by +0.14. On the other hand, evaluation on the test set shows that

the most informative gaze features are from the *context* group learned as multiple auxiliary tasks (`context feats aux`) and they improve the LAS score by +0.18, followed by `fix prob` and $w-1$ `fix prob` with +0.13, `total fix dur` with +0.12 and `late feats aux` with +0.10. Results from both datasets suggest that grouping gaze features and treating them as multiple auxiliary tasks can improve the model's learning.

Discussion The experiments show that our method can give moderate gains for dependency parsing when leveraging gaze information.[6] However, the experiments also show that there is room for improvement, especially coming from generalization capabilities across different treebanks. This also opens the question of whether a different architecture could better suit the purpose of leveraging the gaze information in a consistent way. In this context, a potential line of work could adapt human-attention approaches (Barrett et al., 2018) for structured prediction and word-level classification, although it would come at a cost of speed for parsing as sequence labeling (Strzyz et al., 2019b).

4 Conclusion

This paper has explored how to leverage human data with a competitive dependency parser during training. This contrasts with most of previous work on parsing and gaze annotations, which worked under the assumption that eye-tracking data would be available at inference time. We address this problem and propose a method of leveraging gaze features by using it as an auxiliary task both with and without parallel data. We obtain modest but positive improvements, which opens the question about how to increase the leverage of eye-tracking or other complementary data that is only available during training or comes from a different dataset.

---

[6]Nevertheless, our approach can be employed also to leverage other types of data than eye measurements, i.e. NER or chunking data.

References

Maria Barrett, Željko Agić, and Anders Søgaard. 2015. The Dundee treebank. In *The 14th International Workshop on Treebanks and Linguistic Theories (TLT 14)*.

Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 302–312, Brussels, Belgium. Association for Computational Linguistics.

Maria Barrett, Joachim Bingel, Frank Keller, and Anders Søgaard. 2016. Weakly supervised part-of-speech tagging using eye-tracking data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 579–584, Berlin, Germany. Association for Computational Linguistics.

Maria Barrett and Anders Søgaard. 2015a. Reading behavior predicts syntactic categories. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 345–349, Beijing, China. Association for Computational Linguistics.

Maria Barrett and Anders Søgaard. 2015b. Using reading behavior to predict grammatical functions. In *Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning*, pages 1–5, Lisbon, Portugal. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8. Association for Computational Linguistics.

Susan A Duffy, Robin K Morris, and Keith Rayner. 1988. Lexical ambiguity and fixation times in reading. *Journal of memory and language*, 27(4):429–446.

Evelina Fedorenko, Zachary Mineroff, Matthew Siegelman, and Idan Blank. 2018. Word meanings and sentence structure recruit the same set of fronto-temporal regions during comprehension. *bioRxiv*, page 477851.

Ana V. González-Garduño and Anders Søgaard. 2018. Learning to predict readability using eye-movement data from natives and learners. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018, Proceedings*, pages 5118–5124.

Ana Valeria González-Garduño and Anders Søgaard. 2017. Using gaze to predict text readability. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 438–443, Copenhagen, Denmark. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Nora Hollenstein, Maria Barrett, Marius Troendle, Francesco Bigiolli, Nicolas Langer, and Ce Zhang. 2019. Advancing NLP with cognitive language processing signals. *CoRR*, abs/1904.02682.

Nora Hollenstein and Ce Zhang. 2019. Entity recognition at first sight: Improving NER with eye movement information. 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)*, pages 1–10, Minneapolis, Minnesota. Association for Computational Linguistics.

Alan Kennedy, Robin Hill, and Joël Pynte. 2003. The Dundee corpus. In *Proceedings of the 12th European conference on eye movement*.

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.

Daniel J. Liebling and Sören Preibusch. 2014. Privacy considerations for a pervasive eye tracking world. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, pages 1169–1177, New York, NY, USA. ACM.

Alessandro Lopopolo, Stefan L. Frank, Antal van den Bosch, and Roel Willems. 2019. Dependency parsing with your eyes: Dependency structure predicts eye regressions during reading. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 77–85, Minneapolis, Minnesota. Association for Computational Linguistics.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Mattias Nilsson and Joakim Nivre. 2009. Learning where to look: Modeling eye movements in

reading. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 93–101, Boulder, Colorado. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, 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)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

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 (EMNLP)*, pages 338–348, Copenhagen, Denmark.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681.

Abhinav Deep Singh, Poojan Mehta, Samar Husain, and Rajkumar Rajakrishnan. 2016. Quantifying sentence complexity based on eye-tracking measures. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 202–212, Osaka, Japan. The COLING 2016 Organizing Committee.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019a. Sequence labeling parsing by learning across representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5350–5357, Florence, Italy. Association for Computational Linguistics.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019b. Viable dependency parsing as sequence labeling. 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)*, pages 717–723, Minneapolis, Minnesota. Association for Computational Linguistics.

Takenobu Tokunaga, Hitoshi Nishikawa, and Tomoya Iwakura. 2017. An eye-tracking study of named entity annotation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 758–764, Varna, Bulgaria. INCOMA Ltd.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259.

David Vilares, Mostafa Abdou, and Anders Søgaard. 2019. Better, faster, stronger sequence tagging constituent parsers. 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)*, pages 3372–3383, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA. Association for Computational Linguistics.