Another Dead End for Morphological Tags? Perturbed Inputs and Parsing

Alberto Muñoz-Ortiz and David Vilares
Universidade da Coruña, CITIC
Departamento de Ciencias de la Computación y Tecnologías de la Información
Campus de Elviña s/n, 15071
A Coruña, Spain
{alberto.munoz.ortiz, david.vilares}@udc.es

Abstract

The usefulness of part-of-speech tags for parsing has been heavily questioned due to the success of word-contextualized parsers. Yet, most studies are limited to coarse-grained tags and high quality written content; while we know little about their influence when it comes to models in production that face lexical errors. We expand these setups and design an adversarial attack to verify if the use of morphological information by parsers: (i) contributes to error propagation or (ii) if on the other hand it can play a role to correct mistakes that word-only neural parsers make. The results on 14 diverse UD treebanks show that under such attacks, for transition- and graph-based models their use contributes to degrade the performance even faster, while for the (lower-performing) sequence labeling parsers they are helpful. We also show that if morphological tags were utopically robust against lexical perturbations, they would be able to correct parsing mistakes.

1 Introduction

The use of morphological tags was a core component of dependency parsers to improve performance (Ballesteros and Nivre, 2012). With the rise of neural models, feeding explicit morphological information is a practice that has greatly vanished, with (often) the exception of part-of-speech (PoS) tags. In this line, Ballesteros et al. (2015) already found that character-based word vectors helped improving performance over purely word-level models, specially for rich-resource languages, for which the use of morphological information is more relevant (Dehouck and Denis, 2018). Related, Dozat et al. (2017) showed that predicted PoS tags still improved the performance of their graph-based parser, even when used together with character-based representations. Smith et al. (2018) and de Lhoneux et al. (2017) studied the impact that ignoring PoS tag vectors had on the performance of a biLSTM transition-based parser (Kiperwasser and Goldberg, 2016). They conclude that when considering PoS tags, word-level, and character-level embedddings, any two of those vectors are enough to maximize a parser performance, i.e., PoS tag vectors can be excluded when using *both* word-level and character-level vectors. Zhou et al. (2020) showed the utility of PoS tags when learned jointly with parsing. Recently, Anderson and Gómez-Rodríguez (2021) and Anderson et al. (2021) have explored the differences between using gold and predicted PoS tags, showing that the former are helpful to improve the results, while the latter are often not, with the exception of low-resource languages, where they obtain small but consistent improvements. Furthermore, Muñoz-Ortiz et al. (2022) showed that the efficacy of PoS tags in the context of sequence labeling parsing is greatly influenced by the chosen linearization method.

However, most of such work has focused on: (i) studying the effect of the universal PoS tags (Zeman et al., 2021), and (ii) its impact on non-perturbed inputs. Yet, NLP models are very sensible and brittle against small attacks, and simple perturbations like misspellings can greatly reduce performance (Ebrahimi et al., 2018; Alzantot et al., 2018). This has been shown for tasks such as named-entity recognition, question answering, semantic similarity, and sentiment analysis (Moradi and Samwald, 2021). In parallel, defensive strategies have been tested to improve the robustness of NLP systems, e.g., placing a word recognition module before downstream classifiers (Pruthi et al., 2019), or using spelling checks and adversarial training (Li et al., 2019). Yet, as far as we know, no related work has been done on testing perturbed inputs for parsing and the effect, positive or negative, that using morphological information as explicit signals during inference might have in guiding the parsers.[1]

[1] The code related to this work is available at https://github.com/amunozo/parsing_perturbations.

Findings of the Association for Computational Linguistics: ACL 2023, pages 7301–7310
July 9-14, 2023 ©2023 Association for Computational Linguistics

2 Adversarial framework

Perturbed inputs occur for several reasons, such as for instance on-purpose adversarial attacks (Liang et al., 2018) or, more likely, unintended mistakes made by human writers. In any case, they have an undesirable effect on NLP tools, including parsers. Our goal is to test if under such adversarial setups, coarse- and fine-grained morphological tags: (i) could help obtaining more robust and better results in comparison to word-only parsers (going against the current trend of removing any explicit linguistic input from parsers); or (ii) if on the contrary they contribute to degrade parsing performance.

Below, we describe both how we generate (i, §2.1) linguistically-inspired attacks at character-level, and (ii, §2.2) the tested parsers.

2.1 Perturbed inputs

To perturb our inputs, we use a combination of four adversarial misspellings, inspired by Pruthi et al. (2019) who designed their method relying on previous psycholinguistic studies (Davis, 2003; Rawlinson, 1976). In particular, we consider to: (i) drop one character, (ii) swap two contiguous characters, (iii) add one character, and (iv) replace a character with an adjacent character in a QWERTY keyboard. These changes will probably transform most words into out-of-vocabulary terms, although some perturbations could generate valid tokens (likely occurring in an invalid context). We only apply perturbations to a fraction of the content words of a sentence[2] (details in §3), as function words tend to be shorter and a perturbation could make them unrecognizable, which is not our aim.

Finally, we only allow a word to suffer a single attack. Since we will be evaluating on a multilingual setup, we considered language-specific keyboards to generate the perturbations. We restrict our analysis to languages that use the Latin alphabet, but our adversarial attack would be, in principle, applicable to any alphabetic script.

2.2 Parsing models

Since we want a thorough picture of the impact of using morphological information on parsers, we include three models from different paradigms:

1. A left-to-right transition-based parser with pointer networks (Fernández-González and Gómez-Rodríguez, 2019). It uses biLSTMs (Hochreiter and Schmidhuber, 1997) to contextualize the words, and the outputs are then fed to a pointer network (Vinyals et al., 2015), which keeps a stack and, in a left-to-right fashion, decides for each token its head.

2. A biaffine graph-based parser (Dozat et al., 2017). This model also uses biLSTMs to first contextualize the input sentence. Differently from Fernández-González and Gómez-Rodríguez, the tree is predicted through a biaffine attention module, and to ensure well-formed trees it uses either the Eisner (1996) or Chu (1965); Edmonds (1968) algorithms.[3]

3. A sequence labeling parser (Strzyz et al., 2020) that uses a 2-planar bracketing encoding to linearize the trees. Like the two other parsers, it uses biLSTMs to contextualize sentences, but it does not use any mechanism on top of their outputs (such as biaffine attention or a decoder module) to predict the tree (which is rebuilt from a sequence of labels).

Particularly, we use this third model to: (i) estimate how sensitive raw biLSTMs are to attacks, (ii) compare their behavior against the transition- and graph-based models and the extra mechanisms that they incorporate, (iii) and verify if such mechanisms play a role against perturbed inputs.

Inputs We concatenate a word vector, a second word vector computed at character level, and (optionally) a morphological vector. This is the preferred input setup of previous work on PoS tagging plus its utility for neural UD parsing (de Lhoneux et al., 2017; Anderson and Gómez-Rodríguez, 2021).[4] Note that character-level vectors should be robust against our attacks, but it is known that in practice they are fragile (Pruthi et al., 2019). In this respect, our models use techniques to strengthen their behaviour against word variation, by using character-level dropout. This way, we inject noise during training and give all our models a lexical-level defensive mechanism to deal with misspellings. We kept this feature to keep the setup realistic, as character-level dropout is implemented

[2] Those which universal PoS tags is `ADJ`, `ADV`, `INTJ`, `PROPN`, `NOUN` or `VERB`.

[3] This is true for the `supar` implementation that we use, although Dozat et al. relied on heuristics.

[4] Some authors (Zhou et al., 2020) exploit PoS tags for parsing in a multi-task learning setup instead, but the differences in the experiments are small ($\sim$0.3 points) and they are limited to English and Chinese on non-UD treebanks.

by default in most of modern parsers, and ensure stronger baselines.

Training and hyperparameters We use non-perturbed training and development sets,[5] since our aim is to see how parsers trained in a standard way (and that may use explicit morphological features) behave in production under adversarial attacks. Alternatively, we could design additional techniques to protect the parsers against such perturbations, but this is out of the scope of this paper (and for standard defensive strategies, we already have character-level dropout). For all parsers, we use the default configuration specified in the corresponding repositories. We use 2 GeForce RTX 3090 for training the models for around 120 hours.

Morphological tags To predict them, we use a sequence labeling model with the same architecture than the one used for the sequence labeling parser. We use as input a concatenation of a word embedding and a character-level LSTM vector.

3 Experiments

We now describe our experimental setup:

Data We selected 14 UD treebanks (Zeman et al., 2021) that use the Latin alphabet and are annotated with universal PoS tags (UPOS), language-specific PoS tags (XPOS), and morphological feats (FEATS). It is a diverse sample that considers different language families and amounts of data, whose details are shown in Table 1. For the pre-trained word vectors, we rely on Bojanowski et al. (2017).[6] Also, note that we only perturb the test inputs. Thus, when the input is highly perturbed, the model will mostly depend on the character representations, and if used, the morphological tags fed to it.

Generating perturbed treebanks For each test set, we create several versions with increasing percentages of perturbed content words (from 0% to 100%, with steps of 10 percent points) to monitor how the magnitude of the attacks affects the results. For each targeted word, one of the four proposed perturbations is applied randomly. To control for randomness, each model is tested against 10 perturbed test sets with the same level of perturbation. To check that the scores were similar across runs, we computed the average scores and the standard deviation (most of them exhibiting low values).

Treebank	# Sent.	Family	#UPOS	#XPOS	#FEATS
Afrikaans$_{\text{AfriBooms}}$	1 315	Germanic (IE)	16	95	55
Basque$_{\text{BDT}}$	5 396	Basque	16	-	573
English$_{\text{EWT}}$	12 543	Germanic (IE)	18	51	153
Finnish$_{\text{TDT}}$	12 217	Uralic	16	14	1 786
German$_{\text{GSD}}$	13 814	Germanic (IE)	17	52	458
Hungarian$_{\text{Szeged}}$	449	Uralic	16	-	384
Indonesian$_{\text{GSD}}$	4 477	Austronesian	18	45	48
Irish$_{\text{IDT}}$	4 005	Celtic (IE)	17	72	653
Lithuanian$_{\text{HSE}}$	153	Baltic (IE)	16	30	215
Maltese$_{\text{MUDT}}$	1 123	Afro-Asiatic	17	47	-
Polish$_{\text{LFG}}$	13 774	Slavic (IE)	15	623	1 037
Spanish$_{\text{AnCora}}$	14 305	Latin (IE)	18	318	243
Swedish$_{\text{LinES}}$	3 176	Germanic (IE)	17	214	171
Turkish$_{\text{Penn}}$	14 851	Turkic	15	-	490

Table 1: Relevant information for the treebanks used.

Setup For each parser we trained four models: a word-only (`word`) baseline where the input is just the concatenation of a pre-trained word vector and a character-level vector, and *three* extra models that use universal PoS tags (`word+UPOS`), language-specific PoS tags (`word+XPOS`), or feats (`word+FEATS`). For parsing evaluation, we use labeled attachment scores (LAS). For the taggers, we report accuracy. We evaluate the models on two setups regarding the prediction of morphological tags: (i) tags predicted on the same perturbed inputs as the dependency tree, and (ii) tags predicted on non-perturbed inputs. Specifically, the aim of setup ii is to simulate the impact of using a tagger that is very robust against lexical perturbations.

3.1 Results

Tables 2 and 3 show the average LAS results across all treebanks and models for tags predicted on perturbed and non-perturbed inputs, respectively. Figures 1, 2, and 3 display the mean LAS difference between the `word` and the other model configurations, using tags predicted on both perturbed and non-perturbed inputs for each parser.

3.1.1 Results using morphological tags predicted on perturbed inputs

Figure **??**.a shows the score differences for the transition-based parsers. The average difference between the baseline and all the models using morphological tags becomes more negative as the per-

[5] For the models that use morphological information we went for gold tags for training. The potential advantages of training with predicted PoS tags vanish here, as the error distribution for PoS tags would be different for non-perturbed (during training) *versus* perturbed inputs (during testing).

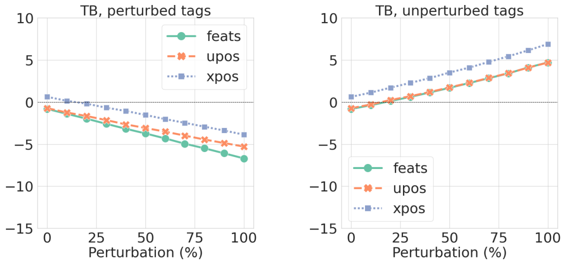
[6] We exclude experiments with BERT-based models for a few reasons: (i) to be homogeneous with previous setups (e.g. Smith et al. (2018), Anderson et al. (2021)), (ii) because the chosen parsers already obtain competitive results without the need of these models, and (iii) for a better understanding of the results, since it is hard to interpret the performances of individual languages while not extracting conclusions biased on the language model used, instead of the parsing architecture.

% Perturbed	Transition-based				Graph-based				Sequence labeling				Tagger accuracy		
	word	UPOS	XPOS	FEATS	word	UPOS	XPOS	FEATS	word	UPOS	XPOS	FEATS	UPOS	XPOS	FEATS
0	75.66	74.93	76.28	74.84	79.35	77.44	78.38	77.28	68.29	68.98	70.96	66.79	89.76	87.80	83.38
10	74.93	73.68	75.07	73.53	78.59	75.69	76.77	75.49	66.71	67.31	69.34	64.97	88.56	86.17	81.68
20	74.11	72.45	73.92	72.13	77.81	73.93	75320	73.73	65.18	65.61	67.76	63.16	87.38	84.59	79.94
30	73.33	71.19	72.66	70.74	76.99	72.22	73.56	71.92	63.62	63.96	66.17	61.37	86.17	82.91	78.22
40	72.52	69.86	71.45	69.33	76.10	70.36	71.88	70.06	62.09	62.24	64.59	59.55	84.93	81.30	76.50
50	71.66	68.58	70.13	67.93	75.27	68.63	70.14	68.09	60.52	60.50	62.94	57.81	83.71	79.61	74.68
60	70.78	67.26	68.75	66.46	74.37	66.72	68.37	66.09	58.94	58.91	61.36	56.10	82.48	77.90	72.92
70	69.87	65.88	67.40	64.92	73.49	64.96	66.64	66.06	57.44	57.24	59.77	54.36	81.19	76.13	71.13
80	68.96	64.50	66.03	63.46	72.48	63.05	64.80	62.27	55.90	55.61	58.17	52.65	79.93	74.42	69.37
90	67.99	63.12	64.61	61.90	71.57	61.12	62.97	60.16	54.42	53.95	56.54	50.96	78.62	72.64	67.56
100	67.04	61.74	63.16	60.34	70.59	59.23	61.14	58.13	52.92	52.30	54.97	49.23	77.30	70.85	65.74

Table 2: On the left, average LAS scores for all treebanks and degrees of perturbation for the word, word+UPOS, word+XPOS, and word+FEATS models *using morphological tags predicted on perturbed input*. On the right, the average scores for the taggers used.

% Perturbed	Transition-based				Graph-based				Sequence labeling			
	word	UPOS	XPOS	FEATS	word	UPOS	XPOS	FEATS	word	UPOS	XPOS	
0	75.66	74.93	76.28	74.84	79.35	77.44	78.38	77.28	68.29	68.98	70.96	66.79
10	74.93	74.64	76.05	74.55	78.59	76.91	78.01	76.78	66.71	68.60	70.53	66.19
20	74.11	74.36	75.82	74.23	77.81	76.46	77.58	73.62	65.18	68.19	70.08	65.62
30	73.33	74.02	75.60	73.94	76.99	75.88	77.20	75.82	63.62	67.76	69.62	64.99
40	72.52	73.71	75.36	73.66	76.10	75.44	76.78	75.27	62.09	67.34	69.13	64.46
50	71.66	73.41	75.17	73.35	75.27	74.94	76.42	74.80	60.52	66.88	68.66	63.79
60	70.78	73.06	74.87	73.04	74.37	74.46	76.02	74.25	58.94	66.40	68.19	63.18
70	69.87	72.74	74.64	72.70	73.49	73.99	75.53	73.76	57.44	65.95	67.72	62.56
80	69.86	72.39	74.40	72.37	72.48	73.46	75.13	73.26	55.90	65.45	67.23	61.92
90	67.99	72.08	74.13	72.10	71.57	72.92	74.46	72.73	54.42	64.93	66.75	61.27
100	67.04	71.73	73.93	71.74	70.59	72.45	74.35	72.15	52.92	64.41	66.27	60.63

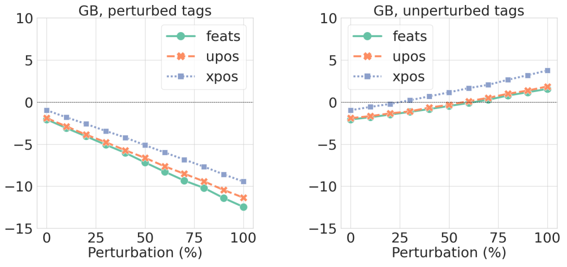
Table 3: Average LAS scores for all treebanks and degrees of perturbation for the word, word+UPOS, word+XPOS, and word+FEATS models *using morphological tags predicted on non-perturbed input*.



(a) Perturbed

(b) Non-perturbed

Figure 1: Average ΔLAS across all treebanks for the transition-based models word+upos, word+xpos, and word+feats vs word, using morphological tags predicted on perturbed and non-perturbed inputs.



(a) Perturbed

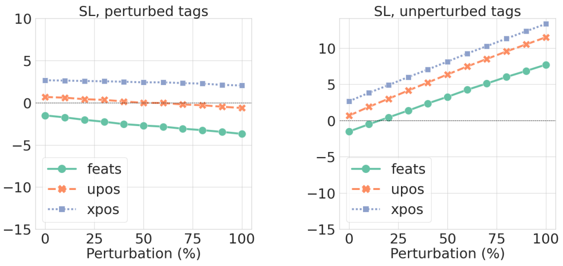
(b) Non-perturbed

Figure 2: Average ΔLAS across all treebanks for the graph-based models word+upos, word+xpos, and word+feats vs word, using morphological tags predicted on perturbed and non-perturbed inputs.

centage of perturbed words increases. Such difference is only positive for word+XPOS when none or a few percentage of words are perturbed. All morphological tags show a similar tendency, word+FEATS degrading the performance the most, followed by the 'coarse-grained' word+UPOS.

Figure 2.a shows the results for the graph-based parsers. Again, most morphological inputs contribute to degrade the performance faster than the baseline. In this case, no model beat the baseline when predicting tags on the perturbed inputs. The performance of word+FEATS and word+UPOS is similar (performing word+UPOS a bit better), and the word+XPOS models improve the performance the most.

Figure 3.a shows the results for the sequence labeling parsers: differences between the baseline and the models utilizing morphological information exhibit minor changes ranging from 0% to 100% of perturbed words. Also, the usefulness of the morphological information depends on the specific tags selected. While word+UPOS obtains similar results to the baseline, word+XPOS scores around 2-3 points higher for the tested percentages of pertur-



(a) Perturbed (b) Non-perturbed

Figure 3: Average ΔLAS across all treebanks for the sequence-labeling models `word+upos`, `word+xpos`, and `word+feats` vs `word`, using morphological tags predicted on perturbed and non-perturbed inputs.

bations, and `word+FEATS` harms the performance in a range between 1 and 4 points.

The results show that feeding morphological tags to both graph- and transition-based parsers has a negative impact to counteract such attacks, degrading their performance faster. On the contrary, the sequence labeling parsers, that rely on biLSTMs to make the predictions, can still benefit from them. In addition, the different trends for the sequence labeling parser *versus* the transition- and graph-based parsers, which additionally include a module to output trees (a pointer network and a biaffine attention, respectively), suggest that such modules are likely to be more effective against adversarial attacks than explicit morphological signals.

3.1.2 Results using morphological tags predicted on non-perturbed inputs

As mentioned above, we use this setup to estimate whether morphological tags could have a positive impact if they were extremely robust against lexical perturbations (see also Figures 1.b, 2.b and 3.b). In the case of the transition-based parser, we observe that morphological tags predicted on non-perturbed inputs help the parser more as the inputs' perturbation grows, being `word+XPOS` the most helpful information, while `UPOS` and `FEATS` become useful only when sentences are perturbed over 20% (but they also become more and more helpful). The graph-based parser also benefits from the use of more precise tags: `word+XPOS` models beat the baseline when the perturbation is over 30%; and over 50% for `word+UPOS` and `word+FEATS` setups. Finally, for the sequence-labeling parser, morphological information from a robust tagger helps the model surpass the baseline for any percentage of perturbed words (except in the case of `word+FEATS`, when it only happens with perturbations over 20%).

3.1.3 Discussion on slightly perturbed inputs

Unintended typos are commonly found among users. For experiments with a small percentage of perturbed words ($< 20\%$), transition-based parsers show improvement solely with the `word+XPOS` model, even when using non-robust taggers. Conversely, graph-based parsers do not benefit from morphological tags in this setup. Last, sequence labeling parsers benefit from incorporating `XPOS` and `UPOS` information, irrespective of the tagger's robustness, but not `FEATS`.

3.1.4 Differences across morphological tags

Averaging across languages, the language-specific `XPOS` tags have a better (or less bad, for setup i) behavior. These tags are specific to each language. The coarse-grained `UPOS` tags have a common annotation schema and tagset. This eases annotation and understanding, but offer less valuable information. For `FEATS`, the annotation schema is common, but in this case they might be too sparse.

4 Conclusion

This paper explored the utility of morphological information to create stronger dependency parsers when these face adversarial attacks at character-level. Experiments over 14 diverse UD treebanks, with different percentages of perturbed inputs, show that using morphological signals help creating more robust sequence labeling parsers, but contribute to a faster degradation of the performance for transition- and graph-based parsers, in comparison to the corresponding word-only models.

Acknowledgements

This paper has received funding from grant SCANNER-UDC (PID2020-113230RB-C21) funded by MCIN/AEI/10.13039/501100011033, the European Research Council (ERC), which has supported this research under the European Union's Horizon Europe research and innovation programme (SALSA, grant agreement No 101100615), Xunta de Galicia (ED431C 2020/11), and Centro de Investigación de Galicia "CITIC", funded by Xunta de Galicia and the European Union (ERDF - Galicia 2014-2020 Program), by grant ED431G 2019/01.

Limitations

Main limitation 1 The experiments of this paper are only done in 14 languages that use the Latin alphabet, and with a high share of Indo-European languages, with up to 4 Germanic languages. This is due to two reasons: (i) the scarcity of XPOS and FEATS annotations in treebanks from other language families, and (ii) the research team involved in this work did not have access to proficient speakers of languages that use other alphabets. Hence, although we created a reasonable diverse sample of treebanks, this is not representative of all human languages.

Main limitation 2 Although we follow previous work to automatically generate perturbations at character-level, and these are inspired in psycholinguistic studies, they might not be coherent with the type of mistakes that a human will make. In this work, generating human errors is not feasible due to the amount of languages involved, and the economic costs of such manual labour. Still, we think the proposed perturbations serve the main purpose: to study how morphological tags can help parsers when these face lexical errors, while the used method builds on top of most of previous work on adversarial attacks at character-level.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Mark Anderson, Mathieu Dehouck, and Carlos Gómez-Rodríguez. 2021. A falta de pan, buenas son tortas: The efficacy of predicted UPOS tags for low resource UD parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 78–83, Online. Association for Computational Linguistics.

Mark Anderson and Carlos Gómez-Rodríguez. 2021. What taggers fail to learn, parsers need the most. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 309–314, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

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*, pages 349–359, Lisbon, Portugal. Association for Computational Linguistics.

Miguel Ballesteros and Joakim Nivre. 2012. MaltOptimizer: A system for MaltParser optimization. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2757–2763, Istanbul, Turkey. European Language Resources Association (ELRA).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Yoeng-Jin Chu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Matt Davis. 2003. Psycholinguistic evidence on scrambled letters in reading.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017. From raw text to Universal Dependencies - look, no tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 207–217, Vancouver, Canada. Association for Computational Linguistics.

Mathieu Dehouck and Pascal Denis. 2018. A framework for understanding the role of morphology in Universal Dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2864–2870, Brussels, Belgium. Association for Computational Linguistics.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Jack Edmonds. 1968. Optimum branchings. *Mathematics and the Decision Sciences, Part*, 1(335-345):25.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Daniel Fernández-González and Carlos Gómez-Rodríguez. 2019. Left-to-right dependency parsing with pointer networks. 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 710–716, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-word applications. In *Network and Distributed Systems Security (NDSS) Symposium 2019*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 4208–4215. AAAI Press.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1558–1570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alberto Muñoz-Ortiz, Mark Anderson, David Vilares, and Carlos Gómez-Rodríguez. 2022. Parsing linearizations appreciate PoS tags - but some are fussy about errors. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 117–127, Online only. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy. Association for Computational Linguistics.

Graham Ernest Rawlinson. 1976. *The significance of letter position in word recognition*. Ph.D. thesis, University of Nottingham.

Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018. An investigation of the interactions between pre-trained word embeddings, character models and POS tags in dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2711–2720, Brussels, Belgium. Association for Computational Linguistics.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2020. Bracketing encodings for 2-planar dependency parsing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2472–2484, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, H͡órunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, Lorena Martín-Rodríguez, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Berzkalne, Manuela Nevaci, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Shafi Sourov, Carolyn Spadine, Rachele Sprugnoli, Steinh͡ór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021. Universal dependencies 2.9. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Houquan Zhou, Yu Zhang, Zhenghua Li, and Min Zhang. 2020. Is pos tagging necessary or even helpful for neural dependency parsing? In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 179–191. Springer.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
5

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3. Experiments

☑ B1. Did you cite the creators of artifacts you used?
3. Experiments

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3. Experiments

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3. Experiments

C ☑ Did you run computational experiments?

2. Adversarial Framework

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
2. Adversarial Framework

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3. Experiments

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3. Experiments

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.