CSSL: Contrastive Self-Supervised Learning for Dependency Parsing on Relatively Free Word Ordered and Morphologically Rich Low Resource Languages

Anonymous ACL submission

Abstract

Neural dependency parsing has achieved remarkable performance for low resource morphologically rich languages. It has also been well-studied that morphologically rich languages exhibit relatively free word order. This prompts a fundamental investigation: *Is there a way to enhance dependency parsing performance, making the model robust to word order variations utilizing the relatively free word order nature of morphologically rich languages?* In this work, we examine the robustness of graph-based parsing architectures on 7 relatively free word order languages. We focus on scrutinizing essential modifications such as data augmentation and the removal of position encoding required to adapt these architectures accordingly. To this end, we propose a contrastive self-supervised learning method to make the model robust to word order variations. Furthermore, our proposed modification demonstrates a substantial average gain of 3.03/2.95 points in 7 relatively free word order languages, as measured by the UAS/LAS Score metric when compared to the best performing baseline.

1 Introduction

Dependency parsing for low-resource languages has greatly benefited from diverse data-driven strategies, including data augmentation (Şahin and Steedman, 2018), multi-task learning (Nguyen and Verspoor, 2018), cross-lingual transfer (Das and Sarkar, 2020), self-training (Rotman and Reichart, 2019; Clark et al., 2018) and pre-training (Sandhan et al., 2021). Further, incorporating morphological knowledge substantially improves the parsing performance for low-resource Morphologically rich languages (MRLs; Dehdari et al., 2011; Vania et al., 2018; Dehouck and Denis, 2018; Krishna et al., 2020a).

MRLs tend to have sentences that follow a relatively free word order (Futrell et al., 2015; Krishna et al., 2020b), as structural information is often encoded using morphological markers rather than word order. In MRLs, a sentence may have different acceptable word order configurations, that preserve the semantic and structural information. However, the permutation invariance is often not reflected in their corresponding semantic space representations encoded using a pretrained model. Pretrained models typically include a position encoding component, often shown to be beneficial for tasks in languages that follow a fixed word order. However, removing the position encoding of the encoder during fine-tuning is demonstrated to be counterproductive (Krishna et al., 2019; Ghosh et al., 2024).

Languages, including MRLs, tend to follow a preferred word order typology. However, such preferences are often followed for the efficiency of communication, from a cognitive, psycho-linguistic, and information-theoretic standpoint and not due to any limitations of the morphology (Krishna et al., 2019; Clark et al., 2023; Xu and Futrell, 2024). For instance, Sanskrit, a classical language (Coulson, 1976), predominantly consists of sentences written as verses in its pre-classic and classic literature. The majority of the available corpora in Sanskrit are written in verse form (Hellwig, 2010–2021). Here, verbal cognition often takes a backseat as words are often reordered to satisfy metrical constraints in prosody (Krishna et al., 2020b, §2). Hence, these sentences appear to be arbitrarily ordered based on syntactic analysis (Kulkarni et al., 2015). In this work, we propose a self-supervised contrastive learning framework, primarily for Sanskrit, that makes the model agnostic to the word order variations within a sentence.

Our Contrastive Self-Supervised Learning (CSSL) framework builds upon the recent success of using annotated pairs in contrastive learning Khosla et al. (2021); Yue et al. (2021) to make the model permutation invariant to the arbitrary word

order variations in Sanskrit (Wright, 1968). Given the comprehensive morphological marking system, the core semantic essence of a sentence remains unaltered, rendering the permuted counterpart as a suitable positive pairing for contrastive learning. This simple use of word permutations in a sentence as positive pairs achieves substantial improvement over prior methods. Our approach, to the best of our knowledge, is the first to use a contrastive learning approach for dependency parsing.

Our proposed approach is modular and agnostic, allowing for seamless integration with any encoder architecture without necessitating alterations to the pertaining decisions. Moreover, our objective is to leverage recent advancements in parsing literature by augmenting with the CSSL framework, which would make these models more robust to word order variations. In this work, we start by examining the robustness of graph-based parsing architectures (Ji et al., 2019; Mohammadshahi and Henderson, 2020, 2021). We believe, graph-based parsing architectures could be a natural choice to model flexible word order. We then focus on investigating essential modifications such as data augmentation Şahin and Steedman (2018) and the removal of position encoding Ghosh et al. (2024) required to adapt these architectures accordingly. We finally show the efficacy of our approach on the best baseline Mohammadshahi and Henderson (2021, RNGTr) model by integrating CSSL with it and report an average performance gain of 3.03/2.95 points (UAS/LAS) improvement over 7 MRLs.

Our main contributions are as follows:

- We propose a novel contrastive self-supervised learning (CSSL) module to make dependency parsing robust for free word order languages.
- Empirical evaluations of CSSL module affirm its efficacy for 7 free word-ordered languages
- We demonstrate statistically significant improvements with an average gain of 3.03/2.95 points over the best baseline on 7 MRLs.

2 Contrastive Self-Supervised Learning

CSSL enables joint learning of representation, via contrastive learning, with the standard classification loss for dependency parsing. Here, via CSSL, we identify sentences which are word-level permutations of each other as similar sentences, and others as dissimilar sentences. The similar sentences are brought closer while pushing dissimilar examples apart (van den Oord et al., 2019; Tian et al., 2020). As shown in Figure 1, the original sentence serves as an anchor point, while its permutations represent positive examples, juxtaposed with randomly generated sentences serving as negative examples. For a given input, when selecting

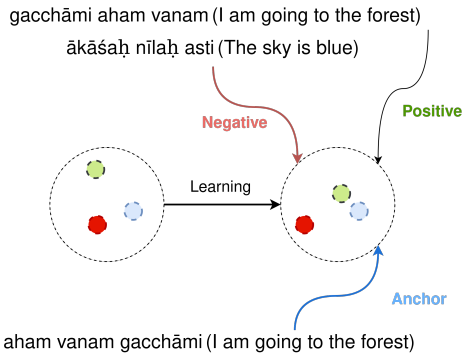

Figure 1: The Contrastive Loss minimizes the distance between an anchor (blue) and a positive (green), both of which have a similar meaning, and maximizes the distance between the anchor and a negative (red) of a different meaning.

a dissimilar sample, we choose a random sentence that clearly differs significantly from any permutation of the given sentence.

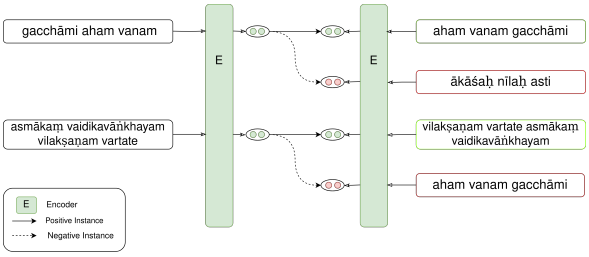

Figure 2: Schematic illustration of the proposed approach for Sanskrit. Self-supervised CSSL leverages the sentence and its permutation pairs as positives and other in-batch instances as negatives.

Formally, as shown in Figure 2 for a sentence X_i (anchor example), its representation should be similar to the permuted instance X_i^+ as permutation [1] does not alter the meaning of a sentence in Sanskrit. However, the representation will differ from a random sentence X_i^- (negative example). Therefore, the distance between the appropriate representations of X_i and X_i^+ is expected to be small. Thus, we can develop a contrastive objective by considering (X_i, X_i^+) a positive pair and $N-1$ negative pairs (X_i, X_i^-) :

[1] Refer to AppendixA.2 for the algorithm to generate the permutations.

$$\mathcal{L}_{\text{CSSL}} = -\log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_{i^+}/\tau\right)}{\sum_{a \in N} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a/\tau\right)}$$

where N represents a batch, z_i represents the representation vector of the anchor sample, z_i^+ denotes the representation vector for the positive sample (permuted sample), z_a represents the representation vector for a sample in the batch (N different samples), and τ is a temperature parameter that controls the concentration of the distribution. For all representation vectors, we employ pooled sentence embedding (Reimers and Gurevych, 2020) for the CSSL loss. Therefore, our final loss is:

$$\mathcal{L} = \mathcal{L}_{\text{CSSL}} + \mathcal{L}_{\text{CE}} \quad (1)$$

The classification loss L_{CE} is the cross-entropy loss applied only to token-level labels of the original training input. The scorer is based on biaffine-scorer (Dozat and Manning, 2017), which tries to independently maximize the local probability of the correct head word for each word.

3 Experiment

3.1 Dataset and metric

As our primary benchmark dataset, we utilize the Sanskrit Treebank Corpus (Kulkarni, 2013, STBC). From STBC, we use a train and dev split of 2,800 and 1,000 respectively. Further, we employ a test set comprising 300 sentences, drawn from the classical Sanskrit work, *Śiśupāla-vadha* (Ryali, 2016).

Moreover, from Universal Dependencies (de Marneffe et al., 2021, UD-2.13), we choose 6 additional morphologically rich low-resource languages, namely, Turkish, Telugu, Gothic, Hungarian, Ancient Hebrew, and Lithuanian.[2] Please note that all the seven languages are chosen from diverse language families and are typologically diverse. Our experiments are primarily focused on a low-resource setting. However, we also show how the framework performs on high-resource MRL. We also experiment with English which is a fixed-ordered high-resource language. Here, we use a training set of 12,544 sentences. We use standard UAS/LAS metrics (McDonald and Nivre, 2011) for evaluation.

Baselines: We utilize Mohammadshahi and Henderson (2020, **G2GTr**), a transition-based dependency parser. Furthermore, we explore Ji et al. (2019, **GNN**) a graph neural network-based model that captures higher-order relations in dependency trees. Finally, we examine Graph-to-Graph Non-Autoregressive Transformer proposed by Mohammadshahi and Henderson (2021, **RNGTr**) which iteratively refines arbitrary graphs through recursive operations.

Model	UAS	LAS
G2GTr (Transition-based)	85.75	82.21
GNN (Graph-based)	88.01	82.8
RNGTr (Graph-based)	89.62	87.43
RNGTr (NoPos)	80.78	78.37
RNGTr (DA)	90.38	88.46
Prop. System (CSSL)	**91.86**	**89.38**
CSSL + DA	92.43	90.18

Table 1: Comparison of graph-based parsers on Sanskrit STBC dataset. We modify the best baseline RNGTr by integrating the proposed method (CSSL) to compare against variants, removing position encoding (NoPos) and data augmentation (DA). The best performances are bold-faced. The results (CSSL vs DA) and (CSSL vs DA+CSSL) are statistically significant as per the t-test with a p-value < 0.01 for the LAS metric.

Hyper-parameters: For RNGTr model, we use the same architecture from the work of Mohammadshahi and Henderson (2020) which uses pre-trained mBERT (Wolf et al., 2020) as the encoder and an MLP and biaffine followed by softmax for the decoder. We adopt the RNGTr codebase with hyperparameter settings as follows: the batch size is 16, the learning rate as 2e-5, the number of transformer blocks as 12 and for the decoder 2 Feed Forward Layers, and the remaining hyperparameters are the same.

3.2 Results

In Table 1, we benchmark graph-based parsers on the Sanskrit STBC dataset. Our proposed contrastive loss module is standalone and could be integrated with any parser.[3] Thus, we modify the best baseline RNGTr by integrating the proposed method (CSSL) and comparing it against variants, removing position encoding (NoPos), and augmenting data augmentation (DA) [4]. Table 1 illustrates that the proposed framework adds a complementary signal making robust word order representa-

[2]The statistics of each of the treebanks used for our experiments is mentioned in Table 4 in the Appendix.

[3]Refer to Appendix A.3 for empirical evidence.

[4]Refer to Appendix A.1 for the algorithm used in Data Augmentation.

Language	Setting	RNGTr		RNGTr + DA		RNGTr + CSSL	
		UAS	LAS	UAS	LAS	UAS	LAS
Turkish-IMST	LRL	72.86	71.99	74.18	72.96	**78.21**	**74.69**
Telugu-MTG	LRL	90.02	80.34	91.86	81.51	**93.79**	**85.67**
Gothic-POIEL	LRL	86.59	81.28	88.61	82.93	**89.15**	**84.19**
Hungarian-SZEGED	LRL	88.13	84.93	90.02	86.65	**91.65**	**87.28**
Ancient Hebrew-PTNK	LRL	90.76	86.42	91.43	87.12	**92.35**	**88.68**
Lithuanian-ALKSNIS	LRL	87.63	83.27	88.41	84.79	**89.82**	**86.45**
Turkish-PENN	HRL	82.31	76.23	85.57	78.19	**88.43**	**80.82**
English-EWT	non-MRL	*92.08*	*90.23*	***93.76***	***92.16***	*93.19*	*90.71*

Table 2: Performance comparison on the RNGTr model on UD Treebanks, RNGTr + DA (Data Augmentation) and RNGTr + CSSL module. The best performances are bold-faced. Our results (CSSL) are statistically significant compared to both RNGTr and RNGTr + DA for each language as per the t-test with a p-value < 0.01 for the LAS metric. LRL stands for low-resource MRL, HRL means high-resource MRL.

tions to RNGTr by improving 2.24/1.95 points in UAS/LAS scores. The performance significantly drops (8.8/9.0 UAS/LAS) when position embeddings are removed (vs. Pos kept) from RNGTr due to train-test mismatch in pretraining and fine-tuning steps. Moreover, our method outperforms data augmentation technique (DA) (Şahin and Steedman, 2018) by 1.48/0.92 points (UAS/LAS) when integrated with the RNGTr baseline. We integrate CSSL on top of an RNGTr+DA system and observe statistically significant improvements of 0.57/0.80 points (UAS/LAS), suggesting the proposed method complements the data-augmentation technique.

Results on multilingual experiments: In this section, we investigate the efficacy of CSSL module in multi-lingual settings. For all MRLs, the trend is similar to what is observed for Sanskrit. Table 2 reports results on 6 other morphologically rich languages in low-resource settings. Our approach averages 3.16/3.12 higher UAS/LAS scores than the usual cross-entropy-based RNGTr baseline. Our system outperforms the rotation-based DA technique with an average increase of 1.74/1.83 in UAS/LAS scores. Here, as expected, our proposed CSSL approach outperforms the standard RNGTr and DA approaches for all the languages, except English. English is not an MRL and it relies heavily on configurational information of the words to understand sentence structure. The DA approach performs better by 0.57/1.45 UAS/LAS scores than our framework. However, it is interesting to note that CSSL still outperforms the RNGTr baseline by 1.11/0.48 UAS/LAS, possibly due to robustness of permutation invariant representation learning we employ in CSSL. As illustrated in Table 1, it is evident that combining CSSL with DA surpasses CSSL alone by approximately 0.5 points, exhibiting a 2-point enhancement over DA.

We also experiment with Turkish on UD_Turkish-PENN Treebank in a high-resource setting, having 14,850 sentences in the training set. Our CSSL framework outperforms usual cross-entropy technique by 6.12/4.59 in UAS/LAS scores and outperforms the DA technique by 2.96/2.63 in UAS/LAS scores. The significant increase in score can be attributed to the greater number of training examples.

4 Conclusion

In this work, we investigated the robustness of graph-based parsing architectures across 7 languages characterized by relatively flexible word order. We introduced a self-supervised contrastive learning module aimed at making encoders insensitive to variations in word order within sentences. Additionally, the modular nature of our approach enables seamless integration with any encoder architecture without necessitating modifications to pretraining decisions. To the best of our knowledge, our approach represents the first utilization of contrastive learning techniques for dependency parsing to address challenges arising from variable word order in low-resource settings. Finally, we demonstrate the effectiveness of our approach by integrating it with the RNGTr architecture Mohammadshahi and Henderson (2021), reporting an average performance improvement of 3.03/2.95 points (UAS/LAS) across the 7 MRLs.

Limitations We could not evaluate on complete UD due to limited available compute resources (single GPU); hence, we selected 7 representative languages for our experiments.

Ethics Statement We do not foresee any ethical concerns with the work presented in this manuscript.

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Thomas Hikaru Clark, Clara Meister, Tiago Pimentel, Michael Hahn, Ryan Cotterell, Richard Futrell, and Roger Levy. 2023. A cross-linguistic pressure for uniform information density in word order.

M. Coulson. 1976. *Sanskrit: An Introduction to the Classical Language*. Teach yourself books. Hodder and Stoughton.

Ayan Das and Sudeshna Sarkar. 2020. A survey of the model transfer approaches to cross-lingual dependency parsing. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(5).

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Jon Dehdari, Lamia Tounsi, and Josef van Genabith. 2011. Morphological features for parsing morphologically-rich languages: A case of Arabic. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 12–21, Dublin, Ireland. 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 and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Quantifying word order freedom in dependency corpora. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100, Uppsala, Sweden. Uppsala University, Uppsala, Sweden.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Poulami Ghosh, Shikhar Vashishth, Raj Dabre, and Pushpak Bhattacharyya. 2024. A morphology-based investigation of positional encodings.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online. Association for Computational Linguistics.

Oliver Hellwig. 2010–2021. Dcs - the digital corpus of sanskrit.

Tao Ji, Yuanbin Wu, and Man Lan. 2019. Graph-based dependency parsing with graph neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2475–2485, Florence, Italy. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2021. Supervised contrastive learning.

Amrith Krishna, Ashim Gupta, Deepak Garasangi, Pavankumar Satuluri, and Pawan Goyal. 2020a. Keep it surprisingly simple: A simple first order graph based parsing model for joint morphosyntactic parsing in Sanskrit. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4791–4797, Online. Association for Computational Linguistics.

Amrith Krishna, Bishal Santra, Ashim Gupta, Pavankumar Satuluri, and Pawan Goyal. 2020b. A graph-based framework for structured prediction tasks in Sanskrit. *Computational Linguistics*, 46(4):785–845.

Amrith Krishna, Vishnu Sharma, Bishal Santra, Aishik Chakraborty, Pavankumar Satuluri, and Pawan Goyal. 2019. Poetry to prose conversion in Sanskrit as a linearisation task: A case for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1160–1166, Florence, Italy. Association for Computational Linguistics.

Amba Kulkarni. 2013. A deterministic dependency parser with dynamic programming for Sanskrit. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 157–166, Prague, Czech Republic. Charles University in Prague, Matfyzpress, Prague, Czech Republic.

Amba Kulkarni, Preethi Shukla, Pavankumar Satuluri, and Devanand Shukl. 2015. How free is free word order in sanskrit. *The Sanskrit Library, USA*, pages 269–304.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1):197–230.

Alireza Mohammadshahi and James Henderson. 2020. Graph-to-graph transformer for transition-based dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3278–3289, Online. Association for Computational Linguistics.

Alireza Mohammadshahi and James Henderson. 2021. Recursive non-autoregressive graph-to-graph transformer for dependency parsing with iterative refinement. *Transactions of the Association for Computational Linguistics*, 9:120–138.

Dat Quoc Nguyen and Karin Verspoor. 2018. An improved neural network model for joint POS tagging and dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 81–91, Brussels, Belgium. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.

Anupama Ryali. 2016. Challenges in developing sanskrit e-readers:semi-automatically using online analyser saṂsĀdhanĪ:with special reference to ŚiŚupĀlavadha of mĀgha. In *Workshop on Bridging 4797the Gap Between Sanskrit CL Tools Management of Sanskrit DL, ICON2016.*

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.

Jivnesh Sandhan, Amrith Krishna, Ashim Gupta, Laxmidhar Behera, and Pawan Goyal. 2021. A little pretraining goes a long way: A case study on dependency parsing task for low-resource morphologically rich languages.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

Clara Vania, Andreas Grivas, and Adam Lopez. 2018. What do character-level models learn about morphology? the case of dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2573–2583, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

J. C. Wright. 1968. J. f. staal: Word order in sanskrit and universal grammar. (foundations of language supplementary series, vol. 5.) xi, 98 pp. dordrecht: D. reidel publishing co., 1967. guilders 30. *Bulletin of the School of Oriental and African Studies*, 31(1):205–205.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation.

Weijie Xu and Richard Futrell. 2024. Syntactic dependency length shaped by strategic memory allocation. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 1–9, St. Julian's, Malta. Association for Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer.

Zhenrui Yue, Bernhard Kratzwald, and Stefan Feuerriegel. 2021. Contrastive domain adaptation for question answering using limited text corpora.

Dejiao Zhang, Shang-Wen Li, Wei Xiao, Henghui Zhu, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2022. Pairwise supervised contrastive learning of sentence representations.

A Appendix

A.1 Data Augmentaion

In our data augmentation (DA) experiments, we employ the *Rotation* algorithm described (Şahin and Steedman, 2018). This approach rearranges the siblings of headwords within a defined set of relations. This alters a collection of words or configuration data, but it does not modify the dependencies.

A.2 Permutation Generation

For generating sentence permutations, we randomly rearrange each word in a sentence to generate phrase permutations while maintaining the relationship between the words. The random permutations are generated while preserving the original dependency tree structures and relations between words in each training sentence. In other words, we first generate the dependency trees for the original sentences and randomly permute the linear order of words, ensuring that the newly permuted sentences still respect the same dependency relations between word pairs.

A.3 Integration of CSSL with another encoder

The modular nature of CSSL framework allows for seamless integration with any encoder architecture, without necessitating alterations to pretraining decisions. We have shown its effectiveness for the best-performing baseline. We are also showing results with one more baseline (for Sanskrit). Our supplementary results indicate that activating contrastive loss for the G2GTr baseline on the STBC treebank for Sanskrit leads to an approximate 2-point enhancement in performance measured by UAS/LAS.

	CE		CSSL	
	UAS	LAS	UAS	LAS
G2GTr	87.16	85.68	89.05	87.05

Table 3: Contrastive Loss with G2GTr on STBC dataset.

A.4 Treebank Statistics

Table 4 provides the detailed statistics for the languages used in the experiments.

A.5 Related Work

Contrastive learning has been the pinnacle of recent successes in sentence representation learning. (Chen et al., 2020) proposed SimCLR by refining the idea of contrastive learning with the help of modern image augmentation techniques to learn robust sets of features. In order to optimize the appropriately designed contrastive loss functions, (Gao et al., 2021; Zhang et al., 2022) uses the entailment sentences in NLI as positive pairs, significantly improving upon the prior state-of-the-art results. To this end, a number of methods have been put forth recently in which the augmentations are obtained through back-translation (Fang et al., 2020), dropout (Yan et al., 2021; Gao et al., 2021), surrounding context sampling (Logeswaran and Lee, 2018; Giorgi et al., 2021), or perturbations carried out at different semantic-level (Wu et al., 2020; Yan et al., 2021).

Treebank	Language Family	train	dev	test
Sanskrit-STBC	Indo-Aryan	2,800	1,000	300
UD-Turkish_IMST	Turkic	3,435	1,100	1,100
UD-Gothic_Proeil	Germanic	3,387	985	1,029
UD-Telugu_MTG	Dravidian	1,051	131	146
UD-Hungarian_Szeged	Uralic	910	441	449
UD-Ancient_Hebrew_PTNK	Semitic	730	439	410
UD-Lithuanian_ALKSNIS	Baltic	2,341	617	684
UD-Turkish_PENN	Turkic	14850	622	924
UD-English_EWT	Roman	12,544	2,001	2,077

Table 4: Treebank Statistics. The number of sentences in train, dev and test for each language.