Version 2024-05-09a

Domain-Weighted Batch Sampling for Neural Dependency Parsing

Jacob Striebel, Daniel Dakota, Sandra Kübler
Indiana University
{jstrieb,ddakota,skuebler}@indiana.edu

1. Summary

- We propose *domain-weighted batch sampling* (DWBS) as a domain adaptation strategy for supervised neural learning.
- We show that DWBS outperforms conventional *randomized batch sampling* (RBS).


```
Algorithm 1 DomainWeightedBatchGenerator
Require: training data set D partitioned into
   subsets D1 and D2
Require: domain-weight parameter μ ∈ [0, 1]
Require: batchSize > 0
  while validation loss continues to decline do
     R1 ← shuffle(copy(D1))
     R2 ← shuffle(copy(D2))
     while True do
        batch ← []
        for n ← 1 to batchSize do
           binaryChoice ← BernoulliSample(μ)
           if binaryChoice = 0 then
              if R1.hasNext() then
                 batch.append(R1.next())
              else
                 break
              end if
           else
              if R2.hasNext() then
                 batch.append(R2.next())
              else
                 break
              end if
           end if
        end for
        if batch.length() < batchSize then
           break
        end if
        yield batch
     end while
  end while
```

2. Domain-Weighted Batch Sampling (DWBS)

- To perform DWBS, before training begins the training data set is partitioned into disjoint *in-domain* and *out-of-domain* subsets.
- The hyperparameter μ is used to define the probability of choosing the next sample from the in-domain subset.
- So, for example, if $\mu = 0.45$, then there is a 45% chance of drawing the next sample from the in-domain subset and a 55% chance of drawing from the out-of-domain subset.

3. Methodology

3.1. Data

- We use Universal Dependencies treebanks version 2.12 (Nivre et al., 2020; de Marneffe et al., 2021), more specifically the English Web Treebank (EWT; Bies et al., 2012) and the Georgetown University Multilayer Corpus (GUM; Zeldes, 2017).
- From the sixteen domains of EWT and GUM, we select only the ten domains that each have a minimum of 1,000 sentences, which includes all five EWT domains and five of the eleven GUM domains.

3.2. Parser

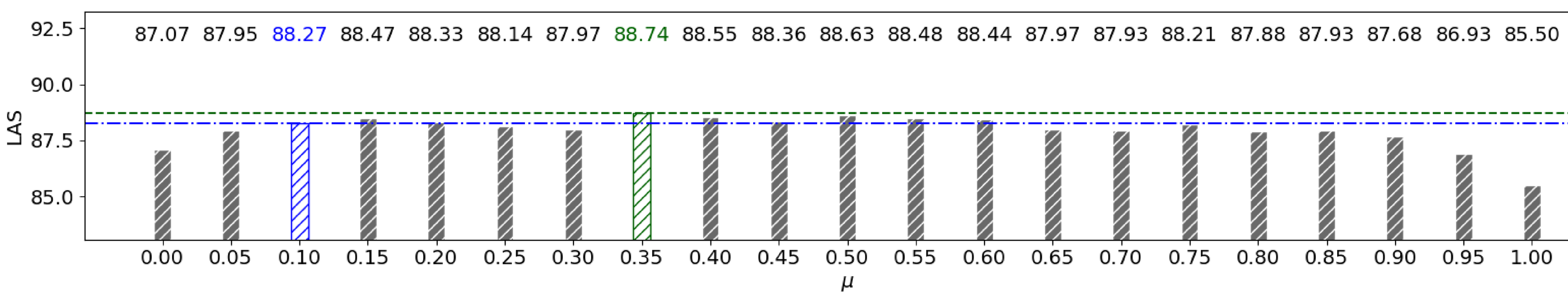
- We use the deep biaffine attention neural dependency parser (Dozat and Manning, 2017) in the implementation by van der Goot et al. (2021).
- We modify the parser so that it can be configured to perform DWBS.
- We use the hyperparameter settings provided by van der Goot et al. with the only modification being that we specify early-stopping patience in terms of batches rather than epochs (see Table 1).

Hyperparameter	Value
Optimizer	Adamw
β_1, β_2	0.9, 0.99
Correction bias	False
Learning rate	0.0001
Weight decay	0.01
Gradient normalization	1
LR scheduler	Slanted triangular
Cut fraction	0.2
Decay factor	0.38
Discriminative fine tuning	True
Gradual unfreezing	True
Batch size	32
Patience batches	200
Max steps	153,600
Embeddings	bert-base-cased
Embeddings dim	768

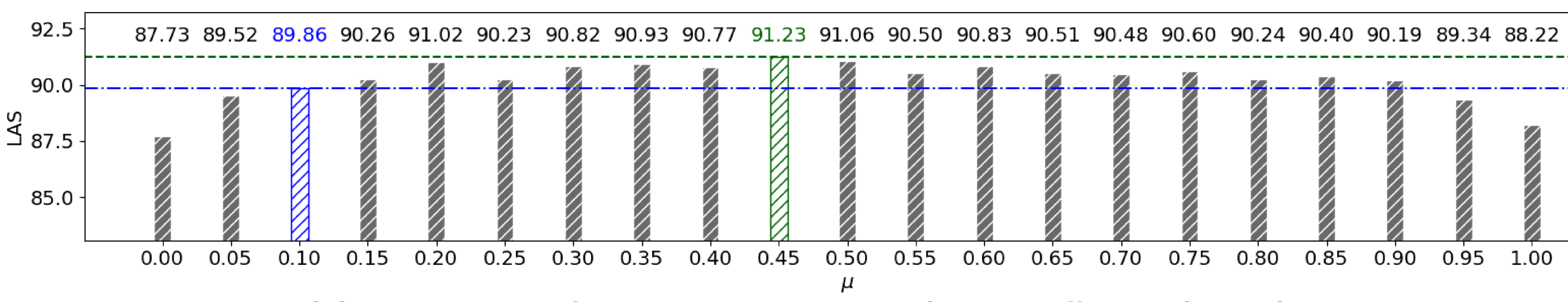
Table 1: Parser hyperparameters

4. Results

(a) Performance of "EWT reviews" parsers



(b) Performance of "GUM fiction" parsers



(c) Parser performance averaged over all ten domains

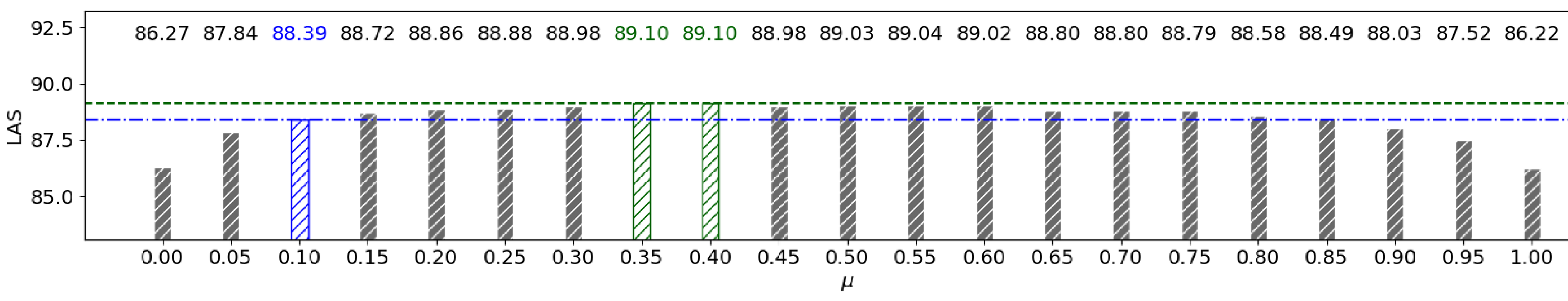


Figure 1: Performance of the DWBS-trained domain-expert parsers on (a) EWT reviews, (b) GUM fiction, and (c) averaged over all ten domains. X-axis: domain-weight hyperparameter μ; y-axis: parser performance in LAS. Because in our experimental setup we use ten domains of equal size, whenever $\mu = 0.10$, DWBS is equivalent to conventional RBS; therefore, in each char we highlight the baseline RBS-trained parser in blue, and we highlight the best performing DWBS-trained parser(s) in green.

4. Results cont.

- In order to evaluate the effectiveness of DWBS, we perform experiments in which we compare a baseline model trained using conventional RBS against domain-expert parsers trained using DWBS.

4.1. Effect on Parsing Accuracy

TB	Domain	μ	LAS R	LAS DW
EWT	Answers	0.35	86.78	87.56
	Email	0.35	86.70	**88.00**
	Newsgr.	0.40	88.64	89.44
	Reviews	0.35	88.27	88.74
	Weblog	0.25	89.52	**90.56**
GUM	Convers.	0.35	85.41	**86.64**
	Fiction	0.45	89.86	**91.23**
	Interv.	0.50	88.08	**89.14**
	Vlog	0.60	87.74	88.57
	Whow	0.35	90.46	91.11

Table 2: Performance in LAS per domain, comparing the baseline parser to the highest-LAS-producing domain-expert parser. LAS R: baseline parser trained using RBS; LAS DW: highest-LAS-producing domain-expert parser trained using DWBS; μ: setting resulting in the highest LAS for the given domain. Improvements of more than 1.00 LAS are bolded.

- The DWBS-trained parser outperforms the baseline in all ten domains tested, for some setting of μ.
- The average improvement across all ten domains, using each domain's best setting of μ, is 0.95 LAS.
- As shown in Table 2, five domains experience gains of more than 1.00 LAS.
- Overall, GUM domains tend to prefer higher values of μ; in other words, those domains profit more from training examples from the same domain, which indicates that each of those domains is different from all others, either in terms of syntactic structures or annotation.

4.2. Effect on Training Duration

Treeb.	Domain	μ	RBS NSC	DWBS NSC	ΔNSC
EWT	Answers	0.35	40.40	40.88	0.48
	Email	0.35	39.84	40.00	0.16
	Newsgroup	0.40	45.60	45.44	-0.16
	Reviews	0.35	40.96	41.36	0.40
	Weblog	0.25	47.04	48.00	0.96
GUM	Conversation	0.35	45.52	41.20	-4.32
	Fiction	0.45	40.56	42.96	2.40
	Interview	0.50	45.20	42.00	-3.20
	Vlog	0.60	48.16	42.96	-5.20
	Whow	0.35	40.40	40.24	-0.16

Table 3: Training duration per domain measured in number of thousands of samples until model convergence, comparing the baseline parser to the highest-LAS-producing domain-expert parser. NSC: number of thousands of training samples until model convergence; RBS NSC: NSC for the baseline parser trained using RBS; DWBS NSC: NSC for the highest-LAS-producing domain-expert parser trained using DWBS; μ: setting yielding the best (in terms of LAS) domain-expert parser for the given domain.

- The domains are evenly split on training time reduction with five seeing a reduction and five experiencing an increase.
- The greatest increase is experience by the GUM fiction domain, which requires 2,400 more sentences than the baseline to achieve convergence.
- The greatest decrease is experienced by the GUM vlog domain, which shows a decrease of 5,200 sentences.
- The average change in training duration is a decrease of 864 sentences.
- The high variability in of differences in training duration may suggest that our target domain data do not always have high internal consistency, which is in line with findings by Zeldes and Schneider (2023), who observed considerable differences in cross-domain parsing between EWT and GUM.

5. Conclusion

- Based on the positive results reported above, when the preconditions for performing DWBS are met, it should be preferred over conventional RBS.

7. References

[1] Bies, Ann and Mott, Justin and Warner, Colin and Kulick, Seth. 2012. English Web Treebank LDC2012T13. Linguistic Data Consortium.
[2] Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In International Conference on Learning Representations.
[3] Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics:System Demonstrations, pages 176–197, Online. Association for Computational Linguistics.
[4] Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. Computational Linguistics, 47(2):255–308.
[5] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In Proceedings of the 12th Language Resources and Evaluation Conference (LREC), pages 4034–4043, Marseille, France.
[6] Amir Zeldes. 2017. Georgetown Multilayer Corpus (GUM). Georgetown University Corpus Linguistics Lab.
[7] Amir Zeldes and Nathan Schneider. 2023. Are UD treebanks getting more consistent? a report card for English UD. In Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023), pages 58–64, Washington, D.C.
[8] Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 1–21, Brussels, Belgium.