Semantic Dependency Parsing for Non-standard Englishes

Suchir Salhan
University of Cambridge
sas245@cam.ac.uk
Word Count: 2856. *

1 Introduction

[1] Despite the increased grammatical and sociolinguistic variation in *Englishes* productively spoken around the world, there is a paucity of high-quality semantic parsers, or parsing techniques, specifically developed for non-standard varieties. The experiments conducted show the benefit of simple data augmentation strategies for improving global model performance for low-resourced varieties of English and highlight the importance of carefully selecting multilingual embeddings and the need to use more complex active learning strategies for example selection.

We first evaluate the performance of a Biaffine Semantic Dependency Parser (Dozat and Manning, 2018) trained on bilexical semantic dependencies from the SemEval 2014 Task 8 (Oepen et al., 2014, 2015) against a novel task formulated by Zhao et al. (2020). This task intrinsically evaluates the ability of semantic parsers to understand the divergence between literal and intended meanings in sentences uttered by L2 English learners. Semantic parsing tools for non-Standard varieties of English have been historically neglected. We perform an error analysis on the frequent errors that occur in the different biaffine models intrinsically evaluated in 2.3.1 against the L1 and L2 test sets of Zhao et al. and further evaluate the biaffine parser on non-standard varieties of English (e.g., code-switching).

2 Modified Bilexical Semantic Parser for L2 English

2.1 Architecture

As a baseline, we train a "vanilla" bilexical semantic dependency parser on the SemEval 2014 Task 8 dataset using the implementation of Candito (2022).[2] The biaffine graph-based architecture, as illustrated in *Figure* 1 for semantic dependency parsing (SDP), introduced by Dozat and Manning (2018), predicts arcs independently from each other – learning to score each possible arc to predict the output structure as a collection of all possible scored arcs. The biaffine parser is a multilayer BiLSTM that creates contextualised representations $c_1, \ldots, c_n = BiLSTM(w_1, \ldots, w_n)$ where w_i is a concatenation of a word embedding, a POS tag embedding, lemma embedding and a character embedding created for the ith token.

$$\mathbf{x}_i = e_i^{(\text{word})} \oplus e_i^{(\text{tag})}$$

The contextualized embeddings are then processed by two feedforward neural networks (FNN), creating specialized representations for potential heads and dependents. The scores for each possible arc-label combination are computed by a final bilinear transformation. The SemEval training corpus is composed of three distinct and parallel semantic dependency annotations (DM, PAS, PSD) of Sections 00-21 of the WSJ Corpus, as well as a balanced sample of twenty files from the Brown Corpus. We only use the training portion of this dataset in the DM schema.

2.2 Modifications and Intrinsic Evaluation

We conduct an empirical evaluation of the biaffine parser on `l1-rerank` and `l2-rerank` datasets produced by Zhao et al. (2020). The implementation of the biaffine parser provided by Candito (2022) does not train properly on 500 sentences, as can be tested using the Colab notebook `DM 500` shared in A. For this reason, all experiments are conducted using Google Colab T4 GPU, with a mini-

[1] Word Count, excluding Appendices, is calculated using `tex.count` on Overleaf

[2] The original code of the ACL 2022 paper can be found here: https://github.com/mcandito/aux-tasks-biaffine-graph-parser-findingsacl22. The code has been modified due to some implementation issues and to adapt the model for the task. All code for the assignment can be found in A.1

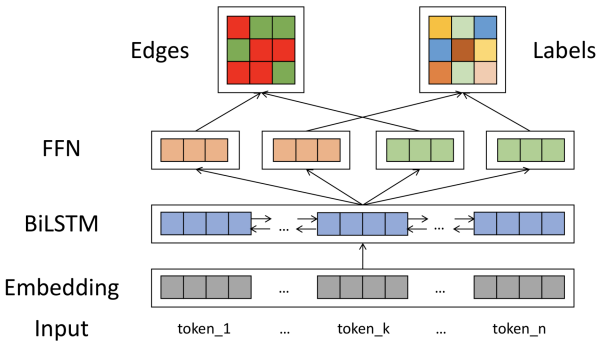

Figure 1: Biaffine Semantic Parser architecture of Dozat and Manning (2018)

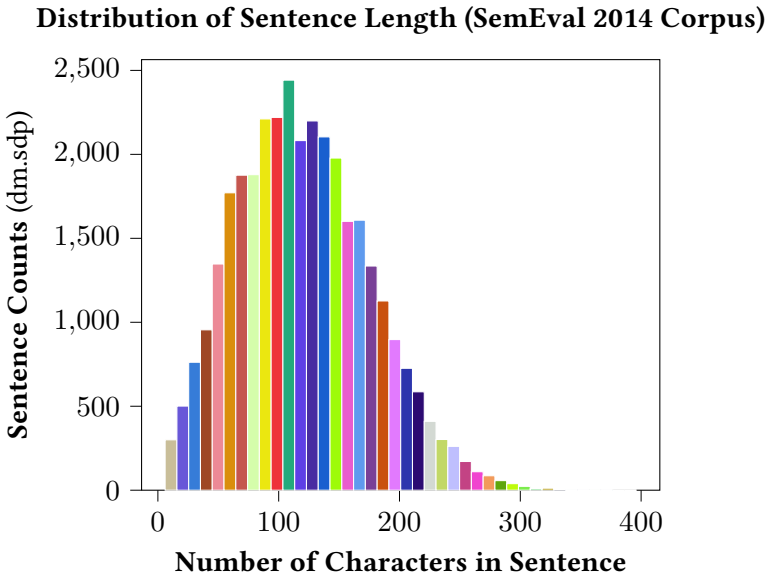

Figure 2: Distribution of length of Sentences in the Training Dataset.

mum training dataset size of 5000 sentences from `dm.sdp`. Labelled and unlabelled F1, precision and recall scores are reported for all three modifications of the "vanilla" biaffine architecture, calculated using a Java toolkit for semantic dependency parsing developed for both the SemEval 2014 and 2015 tasks.[3] Additional scores, including model performance on longer sequences in the test sets, are reported using the test scores outputted from Candito (2022)'s reimplementation of the biaffine parser.

2.2.1 Testing Set from Zhao et al. (2020)

Despite the absence of a gold semantics-annotated corpus for learner English, Zhao et al. exploit English Resource Grammar (ERG) to build a large-scale `sembank` resource with informative semantic representation. They conducting a re-ranking procedure on the K-best candidates derived under the ERG framework with the aid of gold syntactic dependency trees from the Treebank of Learner English (Berzak et al., 2016) to select the semantic graph that best fits the gold syntactic tree T. The SCORE(G_i, T) is a numerical measurement of matching between a semantic graph G_i and T.

$$\hat{G} = \underset{1 \le i \le K}{argmax}\ \text{SCORE}(G_i, T)$$

2.3 Modification 1: A Simple Active Learning Strategy

Active Learning (AL) involves a classifier and an **oracle**– an annotator that cleans, selects and labels the data, feeding it to the parser when required. To implement a naive active learning strategy, we first identity the distribution of sentences in the training data. As illustrated in *Figure* 2, there is an uneven distribution of sentence length in the training set.

To assess the effectiveness of a simple active learning strategies, we order the training sentences according to length. `SMALL` consists of the shortest 5000 sentences in the training dataset, while `LARGE` comprises the longest 5000 sentences. The ten longest and shortest sentences in the training corpus are listed in *Appendix* B, which illustrate a range of sentences approaching 400 characters in length to one-word annotations primarily consisting of Interactive Language, such as exclamatives or single-word utterances. The bilexical semantic dependency framework is ill-suited to the shortest utterances in `SMALL` in *Appendix* B, whose meaning is typically determined by discourse context in the realm of pragmatics.

If a semantic parser is trained exclusively on shorter sequence lengths, as in `SHORT`, it is expected that it will struggle with the longer sentences in the test sets, compared to a model trained on longer text sequences. If a model is trained on smaller sentences, the semantic dependency parser could learn more frequent argument dependencies encoded in bilexical semantics, like proto-roles like Agentive or Patient-like NPs or DPs in transitive sentences or the argument structure properties of intransitive sentences. Of course, these are not explicitly defined in the bilexical dependency formalism. Analysing the effect of sequence size in

[3]The toolkit is available here: https://github.com/semantic-dependency-parsing/toolkit/tree/master

the training data in SMALL and LARGE on model performance allows us to assess the robustness of generalisations the biaffine architecture makes from bilexical semantic dependency annotations.

The results for the overall performance of a model trained on SMALL and LARGE are reported in *Table* 1. The three models are evaluated against the longest sequences in both test sets, as reported in *Table* 2.

2.3.1 Results

30K l1 scores are better than l2 overall, as shown in *Table* 1 , while labelled scores are higher than unlabelled scores, as is consistent with Zhao et al. (2020) initial evaluation. However, LARGE and SMALL are much worse on l1-rerank than l2-rerank. Model performance deteriorates on longer sequences in the test set uniformly across DM 30K, SMALL and LARGE, as shown in a *Table* 2.

2.3.2 A Simple Active Learning Strategy

I use the results to implement a simple active learning strategy, maximising the mixture of sentence lengths in the training data by randomly selecting 2500 sentences from SMALL and LARGE respectively. This is a naive strategy– as discussed in 4.1, several possible extensions could be implemented to refine this. The results in *Table* 3 show that active learning of sentences according to the distribution of sentence lengths in the training corpora does not result in a statistically significant performance improvement compared to DM 5000– a dataset of sentences selected randomly without paying any attention to the role of sentence length. Compared to DM 30K (the full dataset of approximately 32000 sentences), both DM 5000 and MIXED underperform overall. On longer sequences in l1-rerank and l2-rerank, both MIXED and DM 5000 perform far worse than the full training dataset DM 30K, which we discuss in 4.1.

2.4 Modification 2: Data Augmentation

Data augmentation (DA) is a technique that can increase training data diversity by augmenting the training set with "silver data"– slightly modified copies of existing data or synthetic data that act as a regulariser to reduce overfitting and make models more robust (Feng et al., 2021). A heuristic function h transforms a data point and label pair (x, y) to a new augmented sample $(\hat{x}, \hat{y} = h(x, y))$. Ideally, the distribution of augmented data should not be too similar/different from the original data. We experiment with two types of DA techniques.

First, **rule-based DA primitives** have predetermined transforms (*sans*-model-components). Wei and Zou (2019) introduce EDA, which uses simple heuristics like **synonym replacement**, which we use.[4] Other EDA techniques include random insertion/swap/deletion. Second, training data can be augmented using the encoder of a Large Language Model, leveraging contextual word embeddings to find top n similar word for augmentation. This is implemented using the nlpaug package to insert words in sentences by using contextual word embeddings from DistilBERT.[5] While results are only reported for DistilBERT, BERT, RoBERTA or XLNet embeddings can also be used to perform insertion.

I used both methods to generate augmented dictionaries comprised of one augmented sentence for each sentence in SMALL and LARGE. I created a "silver" annotation by retrieving the gold annotation for the sentence in the training data dm.sdp, replacing the second and third columns with the augmented words and lemmas. This was appended to LARGE and SMALL, creating a training set of 10K bilexical semantic dependency annotations.

One minor issue when performing data augmentation is whether the gold POS tags should be preserved or whether the part of speech tags for the augmented sentences should be used, bootstrapped by a POS tagger. After some preliminary experimentation, I use the gold POS tags to perform data augmentation, although this is a method that is possibly more suited to synonym replacement than the second method using contextualised word embeddings.

2.4.1 Results

The two data augmentation strategies implemented lead to roughly comparable performance– simple techniques like synonym selection do not lead to worse model performance than performing word insertion using language model contextualised embeddings. Doubling the size of LARGE and SMALL datasets by additionally producing one augmented sentence for each sentence already in the dataset leads to an overall increase in model performance of $\sim +10$ increase in F1 for SMALL AUGMENTED (compared to SMALL) and $\sim +20$ in-

[4] Code for EDA is sourced from this repository: https://github.com/jasonwei20/eda_nlp

[5] The Python library nlpaug can be found here: https://github.com/makcedward/nlpaug?tab=readme-ov-file

l1-rerank	DM 30K		SMALL		LARGE	
	Unlabelled	Labelled	Unlabelled	Labelled	Unlabelled	Labelled
F1	82.47	75.98	52.31	48.58	56.08	52.16
P	81.14	75.36	63.49	58.97	70.93	65.98
R	81.80	76.60	44.48	41.31	46.36	43.13
l2-rerank	**DM 30K**		**SMALL**		**LARGE**	
	Unlabelled	Labelled	Unlabelled	Labelled	Unlabelled	Labelled
F1	77.81	72.10	50.84	47.19	77.81	72.10
P	77.46	71.78	61.58	57.15	77.46	71.78
R	78.17	72.43	43.30	40.18	78.17	72.43

Table 1: Results for monolingual biaffine semantic parser

l1-rerank	DM 30K		SMALL		LARGE	
	Unlabelled	Labelled	Unlabelled	Labelled	Unlabelled	Labelled
F1	76.02	71.60	28.29	26.09	61.50	57.80
P	82.17	77.39	46.60	42.98	70.68	66.43
R	70.73	66.62	20.31	18.73	54.42	51.15
l2-rerank	**DM 30K**		**SMALL**		**LARGE**	
	Unlabelled	Labelled	Unlabelled	Labelled	Unlabelled	Labelled
F1	69.71	65.62	25.18	22.81	69.71	65.62
P	76.92	72.40	43.04	39.00	76.92	72.40
R	63.75	60.00	17.79	16.12	63.75	60.00

Table 2: Model Performance on long sentences (>40 characters)

crease in F1 for LARGE AUGMENTED (compared to LARGE).

While in Table 1 there was $\sim +20$ difference in F1 scores for LARGE l2-rerank (UL: 77.81, LA: 72.10) compared to l1-rerank (UL: 56.08, LA:72.10), data augmentation appears to be an effective strategy to minimise the performance difference between l1-rerank and l2-rerank. In fact, in both Table 4 and Table 5 the F1 scores for LARGE AUGMENTED are slightly higher for l1-rerank than l2-rerank, as is consistent with the evaluation of `DM 30K` in Table 1.

While the increased size of the dataset using silver data seems to improve the intrinsic performance of the biaffine semantic dependency parser, it does not lead to an improvement on the performance in how models perform on longer test sequences in l1-rerank or l2-rerank. As shown in Table 5, SMALL AUGMENTED has an F1 score of ~ 30 (compared to~ 20 in Table 2 for `SMALL`) on both test sets, while LARGE AUGMENTED has an F1 score of ~ 70 (v ~ 60 on `LARGE`). This suggests that producing "silver data" via data augmentation strategies does not necessarily increase the robustness of models, beyond increasing the quantity of training data.

2.5 Modification 3: Multilingual Embeddings

For non-standard Englishes, we explore the performance benefits of using multilingual mBERT embeddings. The input sentence is segmented by a Wordpiece tokeniser and fed into the mBERT encoder to produce wordpieces. In the mBERT-BiLSTM encoder, the sequence of word embedding vectors is fed to the BiLSTM layer to produce a sentence representation. An initial comparison of multilingual word embeddings is conducted, comparing the mBERT-BiLSTM model with an XLM-BiLSTM model. The results reported in 6 use the `xlm-mlm-tlm-xnli15-1024` model of XLM. Other multilingual encoders can be easily implemented by changing the `-bert name` argument when training the parser. For more information, refer to the code repository shared in A.

2.5.1 Results

The addition of multilingual word embeddings does not lead to an improvement in model per-

l1-rerank	DM 30K		MIXED		DM 5000	
	Unlabelled	Labelled	Unlabelled	Labelled	Unlabelled	Labelled
F1	81.14	75.98	48.89	45.80	49.33	46.42
P	82.47	75.36	70.50	66.05	72.57	68.30
R	81.80	76.60	37.41	35.05	37.36	35.16
F1 (>40 characters)	76.02	71.60	22.67	20.56	27.22	24.94
l2-rerank	**DM 30K**		**MIXED**		**DM 5000**	
	Unlabelled	Labelled	Unlabelled	Labelled	Unlabelled	Labelled
F1	77.81	72.10	47.92	44.42	48.23	45.13
P	77.46	71.78	68.98	63.94	70.80	66.25
R	78.17	72.43	36.71	34.03	36.57	34.22
F1 (>40 characters)	63.75	60.00	18.88	16.70	23.63	21.63

Table 3: Results for monolingual biaffine semantic parser

l1-rerank	SMALL AUGMENTED				LARGE AUGMENTED			
	Synonym		DistilBERT Embedding		Synonym		DistilBERT Embedding	
	UL	LA	UL	LA	UL	LA	UL	LA
F1	61.42	55.82	64.41	59.06	77.65	71.60	77.95	72.01
P	60.04	54.56	69.56	63.78	78.53	72.41	77.95	72.01
R	62.87	57.13	59.97	54.99	76.79	70.81	77.96	72.01
l2-rerank	SMALL AUGMENTED				LARGE AUGMENTED			
	Synonym		DistilBERT Embedding		Synonym		DistilBERT Embedding	
	UL	LA	UL	LA	UL	LA	UL	LA
F1	59.48	53.86	61.94	56.73	73.40	67.59	73.51	67.81
P	58.33	52.8	66.73	61.13	74.71	68.80	74.10	68.35
R	60.67	54.94	57.78	52.93	72.13	66.42	72.93	67.27

Table 4: Data Augmentation Results

formance, although the F1 scores for XLM 30K marginally outperform those of DM 30K. However, in Table 6, the addition of mBERT word embeddings leads to a deterioration of performance on SMALL and LARGE compared to 30K. The addition of multilingual word embeddings leads to a 20 point decrease in F1 scores for SMALL and a significant deterioration of model performance in LARGE, as illustrated in Table 6.

When training mBERT-BiLSTM semantic dependency parser on LARGE, there was a substantial deterioration in Recall. This also occurred when the experiment was replicated. This possibly suggests a general weakness of using multilingual word embeddings to train a semantic parser on complex sentences, which as illustrated in Appendix B can be of the order of close to 400 characters long and contain complex language-specific syntactico-semantic constructions (e.g, quotatives, complementation, negation) that have significantly more complex adjunction and modification structures. In Table 7, the low Recall of the mBERT-BiLSTM trained on LARGE leads to a worse performance on the longest character sequences in both test sets. However, the addition of a multilingual encoder appears to strengthen the improvement of SMALL, with a higher F1 score of close to 30 when trained using a mBERT-BiLSTM architecture compared to a baseline F1 score of around 25.

However, the performance of mBERT embeddings shows a general deterioration of model performance on the skewed distribution sets LARGE and SMALL, despite a comparable global performance of mBERT 30K and XLM 30K to DM 30K. Since the mBERT-BiLSTM does not improve the performance of SMALL or LARGE on `l1-rerank` or `l2-rerank`, the main performance benefit of a multilingual encoder seems to arise on the training data that is neither in SMALL or LARGE. While

l1-rerank	SMALL AUGMENTED				LARGE AUGMENTED			
	Synonym		DistilBERT Embedding		Synonym		DistilBERT Embedding	
	UL	LA	UL	LA	UL	LA	UL	LA
F1	35.00	30.89	37.30	33.67	74.86	69.88	75.85	70.99
P	31.54	27.84	43.21	39.01	73.98	69.06	74.34	69.57
R	39.31	34.69	32.81	29.62	75.77	70.73	77.42	72.46
l2-rerank	SMALL AUGMENTED				LARGE AUGMENTED			
	Synonym		DistilBERT Embedding		Synonym		DistilBERT Embedding	
	UL	LA	UL	LA	UL	LA	UL	LA
F1	32.91	29.30	33.12	29.59	70.85	66.35	71.61	67.20
P	29.91	26.63	40.08	35.80	71.14	66.62	71.59	67.18
R	36.59	32.58	28.23	25.22	70.57	66.09	71.64	67.22

Table 5: Data Augmentation Results on long sequences (>40 characters)

Multilingual word embeddings generally perform well when evaluating the lexical semantics on word similarity tasks, their benefits for semantic parsing are unclear.

3 Error Analysis & Non-Standard English Corpus

I use the `eval.py` script from Candito (2022) to analyse where the models that were intrinsically evaluated falter semantically. All evaluation script files are included in the code repository shared in A. I generate accuracy scores for the bag of labels in the dataset for each dataset intrinsically evaluated. The full set of scores can be found in A.2. The average bag of labels accuracy for `l1-rerank` is 69.04. For `l2-rerank` it is 68.97. There is not a significant difference between both test datasets in accuracy across labels. We can apply non-parametric significance tests, which are standardly used when the test statistic distribution is unknown, to measure whether two independent samples ("L1 sentences" and "L2 sentences") have the same distribution of Semantic Dependency Parsing scores. Two common test are the sign test and the permutation test (Dror et al., 2018). I use the paired permutation test, which checks whether the significance of the test statistic by evaluating the probability that a value at least as large as the observed statistic would occur if system outputs were randomly swapped (Zmigrod et al., 2022). The paired permutation significance testing system can be used off-the-shelf to produce an exact p-value.[6]

Qualitatively, I found that `mBERT Large` substantially underpredicts the ARGs in the dataset. Statistically, the model 302 labels, compared to 8762 Test ARG1 labels; 518 ARG2 labels for 5536 ARG1) in `l1-rerank`. The system outputs zero `then-c` labels, while 2016 labels appear in the Gold. This results in a distribution of labels in the output file that deviates from the distribution of individual labels in the test set. However, neither test dataset contains `top` labels. However, the error analysis files produced show that all parsing systems trained on the `dm.sdp` dataset produce `top` labels – for example, `LARGE` evaluated on `l2`-rerank generates 2124 labels. Due to the frequency of `top` labels in the bilexical semantic dependency formalism, the scores reported for the intrinsic evaluation are lower than the true model performance. There are additionally some annotation errors in the test dataset. For example in `l1-rerank`, there is an `andc` link to *not* from `grew` in the sentence:

`I grew up in a world full of technology and for me the recent development in technology is not very great.`

Second, I evaluate the models on simple Hindi-English code-switched sentences shared from Homework 2.[7] To do this, I manually create a SDP file according to the SemEval annotation standard[8] as illustrated in *Figure* 3. For the English-Hindi

[6] Paired Permutation Test Library available from: https://github.com/rycolab/paired-perm-test

[7] I use Homework 2 sentences shared by another Part III student, which have simpler semantic graphs

[8] For full information about the `DM` annotation format: https://alt.qcri.org/semeval2014/task8/index.php?id=data-and-tools

l1-rerank	mBERT						XLM	
	30K		SMALL		LARGE		30K	
	UL	LA	UL	LA	UL	LA	UL	LA
F1	81.27	75.14	30.45	26.05	37.21	33.86	84.50	77.53
P	80.39	74.32	68.25	58.39	71.19	64.78	81.43	74.72
R	82.18	75.98	19.60	16.77	3.23	2.94	87.80	80.56
l2-rerank	30K		SMALL		LARGE		30K	
	UL	LA	UL	LA	UL	LA	UL	LA
F1	77.16	71.23	30.37	25.88	36.46	34.41	79.98	73.28
P	76.60	70.71	67.62	57.61	69.84	63.45	77.24	70.77
R	77.73	71.76	19.59	16.69	3.09	5.38	82.92	75.98

Table 6: Multilingual Embeddings

l1-rerank	mBERT						XLM	
	30K		SMALL		LARGE		30K	
	UL	LA	UL	LA	UL	LA	UL	LA
F1	76.07	71.54	32.42	27.53	29.85	24.49	79.85	74.08
P	80.15	75.38	55.43	47.06	58.21	47.76	78.30	72.64
R	72.38	68.08	9.42	8.00	1.50	1.23	81.46	75.58
l2-rerank	30K		SMALL		LARGE		30K	
	UL	LA	UL	LA	UL	LA	UL	LA
F1	70.15	67.07	35.22	31.62	36.92	31.89	74.49	68.92
P	75.44	72.13	61.95	55.61	70.97	61.29	73.88	68.36
R	65.55	62.68	8.49	7.63	2.88	2.49	75.12	69.50

Table 7: Results for multilingual embeddings on long sentences (>40 characters).

sentence `students are prone to late` **sona**, I create a DM annotation using "augmented English" – translating the code-switched words/phrases but retaining the word order. Further experimentation is needed to assess the most appropriate preprocessing strategies for low-resource code-switching. mBERT LARGE does not predict a single label correctly. Ensuring to annotate `top` in the test file, all biaffine models tested (mBERT Small/Large, Large, Small, SMALL/LARGE AUGMENTED) correctly label the semantic topic of the translated code-switched utterance.

#Students are prone to being late.

1	Students	Students	NN	-	-	ARG1	_	_	
2	are	are	VBP	+	+	_	_	_	
3	prone	prone	JJ	-	-	_	_	_	
4	to	to	TO	-	+	_	_	_	
5	sleeping	sleeping	VBG	-	-	_	ARG1	_	
6	late	late	RB	-	+	_	_	ARG2	
7	.	.	-	-	-	_	_	_	

#I cannot move this chair further back

1	I	I	PRP	-	-	_	ARG1	_	_
2	can	can	MD	-	+	_	_	_	_
3	not	not	RB	+	-	ARG1	_	_	_
4	move	move	VB	-	+	_	_	_	_
5	this	this	DT	-	+	_	_	_	_
6	chair	chair	NN	-	-	_	_	BV	_
7	further	further	RB	-	+	_	_	_	_
8	back	back	RB	-	-	_	_	_	ARG1
9	.	.	-	-	-	_	_	_	_

Figure 3: Manually creating gold-standard bilexical semantic dependency annotations according to the SemEval 2014 annotation standard of Oepen et al. (2014, 2015)

4 Conclusion and Further Directions

4.1 Further Directions

There are several plausible extensions of the techniques used to modify the classic biaffine semantic dependency parser architecture. More sophisticated Active Learning can enforce diversity in the sampled batches (in this case `LARGE` and `SMALL`), using determinantal point processes (DPPs) to improve over their diversity-agnostic counterparts (Shi et al., 2021). Given the paucity of annotated data for non-standard Englishes, it is expensive to construct gold datasets for code-switched Englishes, even in a comparably light-weight formalism like DM. In addition to developing data

augmentation techniques with a standard preprocessing pipline, semi-supervised SDP models are a useful alternative– capable of learning from both labeled and unlabeled data (Jia et al., 2020).

The extrinsic evaluation of bilexical semantic parsers in natural language semantics tasks is important in measuring the utility of a semantic parser for non-standard Englishes. Analogous to extrinsic tasks for syntactic parsing (Fares et al., 2018), semantic parsers can be evaluated extrinsically on semantic role-labelling, which has been modelled using "higher-order" semantic graphs that extend the bilexical formalism to consider interactions between predicate and argument pairs (Li et al., 2020), or on negation resolution and coreference resolution.

4.2 Conclusion

The experiments conducted here highlight the benefits of data augmentation techniques and the importance of exploring sentence length distribution as a factor modulating intrinsic model performance. The qualitative analysis conducted highlights the challenges of developing high-quality test resources for non-standard Englishes, including code-switched varieties, while the quantitative evaluation suggests that using multilingual embeddings does not necessarily provide a straightforward benefit to semantic parsers. State-of-the-art semantic dependency parsers are data-hungry and producing high-quality annotations for low-resourced and non-standard varieties of English is expensive. The techniques explored here, albeit naive and simple, provide productive future directions to develop more equitable semantic parsing datasets for non-standard varieties and underscore the need to develop new techniques and even model architectures that are not so dependent on vast quantities of annotations that are challenging to curate for low-resourced languages and varieties.

References

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal Dependencies for learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746, Berlin, Germany. Association for Computational Linguistics.

Marie Candito. 2022. Auxiliary tasks to boost biaffine semantic dependency parsing. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2422–2429, Dublin, Ireland. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Murhaf Fares, Stephan Oepen, Lilja Øvrelid, Jari Björne, and Richard Johansson. 2018. The 2018 shared task on extrinsic parser evaluation: On the downstream utility of English Universal Dependency parsers. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 22–33, Brussels, Belgium. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Zixia Jia, Youmi Ma, Jiong Cai, and Kewei Tu. 2020. Semi-supervised semantic dependency parsing using CRF autoencoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6795–6805, Online. Association for Computational Linguistics.

Zuchao Li, Hai Zhao, Rui Wang, and Kevin Parnow. 2020. High-order semantic role labeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1134–1151, Online. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926, Denver, Colorado. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72,

Dublin, Ireland. Association for Computational Linguistics.

Tianze Shi, Adrian Benton, Igor Malioutov, and Ozan İrsoy. 2021. Diversity-aware batch active learning for dependency parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2616–2626, Online. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Yuanyuan Zhao, Weiwei Sun, Junjie Cao, and Xiaojun Wan. 2020. Semantic parsing for English as a second language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6783–6794, Online. Association for Computational Linguistics.

Ran Zmigrod, Tim Vieira, and Ryan Cotterell. 2022. Exact paired-permutation testing for structured test statistics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4894–4902, Seattle, United States. Association for Computational Linguistics.

A Appendix

A.1 Implementation

All the code and datasets used in this dissertation are available in this GitHub repository: `https://github.com/suchirsalhan/L98`, or alternatively here, on Google Drive. This repository contains:

1. A reimplementation of the Biaffine Semantic Dependency Parser of Dozat and Manning (2018). It modifies the code repository of Candito (2022), which contains some minor errors.

2. The datasets generated from the training set and the code to replicate the process of dataset generation for data augmentation.

3. Colab Notebooks to replicate experiments and evaluation of SDP on the test sets

4. The repository contains code-generation datasets and error analysis files.

		Accuracy
30K	11	78.96
30K	12	76.76
SMALL	11	61.88
SMALL	12	61.54
LARGE	11	63.54
LARGE	12	76.76
SMALL AUGMENTED SYNONYM	11	66.5
SMALL AUGMENTED SYNONYM	12	65.66
LARGE AUGMENTED SYNONYM	11	75.85
LARGE AUGMENTED SYNONYM	12	73.69
mBERT 30K	11	78.32
mBERT 30K	12	76.19
XLM 30K	11	81.03
XLM 30K	12	78.34
mBERT SMALL	11	55.86
mBERT SMALL	12	55.69
mBERT LARGE	11	53.58
mBERT LARGE	12	53.33
LARGE AUGMENTED EMBEDDINGS	11	76.07
LARGE AUGMENTED EMBEDDINGS	12	73.79
SMALL AUGMENTED EMBEDDINGS	11	67.85
SMALL AUGMENTED EMBEDDINGS	12	66.99

Figure 4: Accuracy Bag of Labels Scores for all datasets intrinsically evaluated

A.2 Accuracy

A.3 Hyperparameters

We use the final hyperparameter configuration of Dozat and Manning illustrated in 8.

Hidden Layer	**Hidden Sizes**
Word/Glove/POS/Lemma/Char	100
GloVe Linear	125
Char linear	100
Encoder BiLSTM	3 * 600
Arc/Label	600
CharLSTM	1 *400
Dropouts	**Dropout Probability**
Word/Glove/POS/Lemma/Char	20%
Char LSTM (FF/recur)	33%
Char linear	33%
Encoder BiLSTM	45% / 25%
Arc/Label	25% / 33%
Optimiser & Loss	**Value**
Adam β_1	0
Adam β_2	0.95
Learning Rate	$1e^{-3}$
L2 Regularisation	$3e^{-9}$
Interpolation λ	0.025

Table 8: Biaffine Semantic Parser architecture of Dozat and Manning

B Selected Examples from the Training Corpus

I include the ten shortest and five longest sentences in the training dataset dm.sdp to illustrate the diversity of sentences lengths and semantic structures represented in SMALL and LARGE.

B.1 LARGE

396 characters

In this light , the comparative advantages of legislative law-making become clear : (1) Before it acts , the legislature typically will hear the views of representatives of all those affected by its decision , not just the immediate parties before the court ; and (2) the legislature can frame " bright line " standards that create less uncertainty than the fact-bound decisions of courts .

380 characters

The department would be required to block the buy-out if the acquisition is likely to financially weaken a carrier so that safety would be impaired ; its ability to compete would be sharply diminished ; it would be put into foreign control ; or if the transaction would result in the sale of airline-related assets – unless selling such assets had an overriding public benefit .

374 characters

Because many of these subskills – the symmetry of geometrical figures , metric measurement of volume , or pie and bar graphs , for example – are only a small part of the total fifth-grade curriculum , Mr. Kaminski says , the preparation kits would n't replicate too many , if their real intent was general instruction or even general familiarization with test procedures .

369 characters

It is no coincidence that from 1844 to 1914 , when the Bank of England was an independent private bank , the pound was never devalued and payment of gold for pound notes was never suspended , but with the subsequent nationalization of the Bank of England , the pound was devalued with increasing frequency and its use as an international medium of exchange declined.

380 characters

" Our intensive discussions with Jaguar , at their invitation , " GM said , " have as their objectives to create a cooperative business relationship with Jaguar that would provide for the continued independence of this great British car company , to ensure a secure future for its employees and to provide an attractive long-term return for its shareholders . " 363

B.2 SMALL

Sentence ID	Utterance
21591001	Ing.
21778011	Wow!
21778015	JKD:
21778021	HRH:
21778097	KIM:
21778119	HLR:
21778131	FIG:
20052005	TV:
21625003	No?
21778066	RD:

Table 9: Shortest Utterances in dm.sdp