Cross-Lingual Domain Adaptation for Dependency Parsing

Sara Stymne
Department of Linguistics and Philology
Uppsala University
sara.stymne@lingfil.uu.se

Abstract

We show how we can adapt parsing to low-resource domains by combining treebanks across languages for a parser model with treebank embeddings. We demonstrate how we can take advantage of in-domain treebanks from other languages, and show that this is especially useful when only out-of-domain treebanks are available for the target language. The method is also extended to low-resource languages by using out-of-domain treebanks from related languages. Two parameter-free methods for applying treebank embeddings at test time are proposed, which give competitive results to tuned methods when applied to Twitter data and transcribed speech. This gives us a method for selecting treebanks and training a parser targeted at any combination of domain and language.

1 Introduction

Recent advances in dependency parsing have enabled high-quality parsing for a relatively high number of languages. However, satisfactory results are mainly limited to text types for which there are treebanks for a specific language. Even for high-resource languages, treebanks are typically only available for a small number of domains and genres. In this work we show how we can improve parsing for non-canonical text types by using in-domain annotated data from other languages.

We focus on two low-resource text types that stand out in different respects from canonical written texts: Twitter data and (transcribed) spoken data, for which annotated treebanks exist for only a small number of languages. Twitter data often contains non-standard language and specific features such as hash tags and emoticons. Spoken data tends to be more informal than written texts, and contains features such as fillers, restarts, and reparandums. While Twitter can be regarded as a genre, and spoken data as a medium (Lee, 2001), we will follow previous work in NLP and use the term *domain* to cover both these types of text.[1]

The main novelty in this work is that we combine domain adaptation with cross-lingual learning for dependency parsing. We note that treebanks for a specific domain (IND: in-domain) often exist for some languages, and we show that we can take advantage of such data for parsing this domain in other languages. Our main focus is on the case where we want to parse data for a language that has some resources, but none for the domain in question (OOD: out-of-domain). While there is plenty of work both on cross-lingual parsing (Ammar et al., 2016a; Ahmad et al., 2019; Kondratyuk and Straka, 2019) and domain adaptation for parsing (Kim et al., 2016; Sato et al., 2017; Xiuming et al., 2019), there is to the best of our knowledge no attempts to combine these approaches in a uniform framework for dependency parsing.

We adapt the parsing framework of Smith et al. (2018a) which incorporates treebank embeddings to represent treebanks, similarly to how language embeddings has been used to represent the languages (Ammar et al., 2016b; de Lhoneux et al., 2017a). In this framework each parsing model is trained on a concatenation of different treebanks, and the representation of each input token includes an embedding

[1] The term *domain* has often been used as a catch-all term in NLP, to cover many different types of text type differences, often without being clearly defined, see e.g. (Weiss et al., 2016; Chu and Wang, 2018), even though there has been some attempts to investigate different aspects of *domains*, e.g. (van der Wees et al., 2015; Ruder et al., 2016).

representing the treebank from which the token comes from. Depending on the mix of treebanks, the treebank embedding can encode aspects such as differences between languages, domains, and annotation style. Parsing with treebank embeddings has previously been applied monolingually (Stymne et al., 2018; Wagner et al., 2020) and cross-lingually for related languages, but without taking domain into account (Smith et al., 2018a; Lim et al., 2018),[2] In this paper, we show that joint training with treebank embeddings can be applied simultaneously across both across languages and domains, in effect addressing the task of cross-lingual domain adaptation. It is a simple and efficient method, which does not require expensive pre-processing, pre-training, translation, or similar tasks required by many other cross-lingual approaches, while giving competitive results across many settings. In this work we explore how such a resource lean method can be applied to cross-domain parsing on its own. We leave to future work an investigation of how the proposed technique interacts with other techniques for domain adaptation, for instance based on pre-training contextualized embeddings like BERT (Devlin et al., 2019).

At test time, there is a need to determine which treebank embedding to use, which is straightforward for test data from a treebank used during training. However, when the input sentence is from a treebank not used during training there is a need to determine the treebank embedding. One option is to use a *proxy* treebank (Stymne et al., 2018), i.e. to choose the embedding of one of the treebanks used during training, which can be determined based on development data. Wagner et al. (2020) show that it is often advantageous to interpolate the embeddings of the treebanks used for training instead. They show in a monolingual setting how interpolation weights can be learnt based on sentence similarity. However, their equal weight baseline performs just as well in the majority of cases, and avoids the need of learning interpolation weights, which would also be less straight-forward in the cross-lingual setting. We thus adopt equal-weight interpolation. We also propose the use of an ensembling strategy applied to trees obtained by using all possible proxy treebanks embeddings.

We show that using in-domain data from another language is useful when no in-domain data is available for the target language. Using the proposed methods, we can potentially train a parser for any combination of domain and language, as long as that domain has training data in some language, without the need for tuning on target development data.

2 Experimental Setup

Data We mainly use data from the Universal Dependencies (UD) project (Nivre et al., 2020), version 2.4 (Nivre et al., 2019). We put our main focus on languages with a single-domain dependency treebank with either spoken data or Twitter data, including both training and test data and additional treebank data for other domains. While several UD treebanks contain some data from these domains mixed with other domains, it is often not easily identifiable which domain specific sentences come from. We thus use the three UD single domain treebanks of spoken data for French, Norwegian, and Slovenian, which fulfills our requirements. In addition we evaluate our methods on Komi-Zyrian and Naija, which both have spoken test data, but no training data for any domain in UD. For Twitter we use two treebanks from UD for Italian and code-switching Hindi–English. In addition we use the English Tweebank v2, which is annotated in UD style (Liu et al., 2018). We convert sentences in the English Tweebank with multiple roots to have only one root, which is a UD requirement, by only keeping the first root, and joining the other roots to it with the *parataxis* relation. This happens when a single Tweet contains more than one sentence, and it is the solution adopted in the Italian PoSTWITA treebank.

In addition to the in-domain treebanks we use additional treebanks from the same language, when available, or for related languages otherwise. For Komi Zyrian, a Uralic lanugage, we also use a Russian treebank, since Russian is a contact language, which also shares the Cyrillic script, in contrast to other Uralic treebanks with training data. Table 1 lists the data used for each language. Note that in all cases, the additional data is much larger than the in-domain data, which is typically quite small. For Slovenian SST, no development data was available, so we split off 5% of the training data. In all other cases we use the original splits. While UD treebanks have standard annotation guidelines, there are several inconsistencies

[2] With the exception of a footnote in Smith et al. (2018a), where this type of data combination is mentioned for spoken French and Naija. However, no details or experimental results are provided.

Language	IND Treebank	Train	Dev	Test	Additional OOD data
French	Spoken	15.0K	10.2K	10.2K	GSD (364K), *Partut* (24.9K), Sequoia (51.9K)
Norwegian	NynorskLIA	35.2K	10.2K	10.0K	*Nynorsk* (245K), Bokmaal (244K)
Slovenian	SSJ	18.6K	906	10.0K	*SST* (113K), Croatian_SET (153K), Serbian_SET (74.3K)
Komi Zyrian	IKDP	–	–	1.3K	Finnish_TDT (163K), North_Sami_Giella (16.8K), Russian_Taiga (18.1K)
Naija	NSC	–	–	12.9K	English: EWT (205K), GUM (66.2K), LinES (50.1K) ParTUT (43.5K)
English	Tweebank	24.8K	11.8K	19.1K	EWT (205K), GUM (66.2K), LinES (50.1K) *ParTUT* (43.5K)
Hindi–English CS	HIENCS	19.3K	3.3K	3.1K	English: EWT (205K), GUM (66.2K), LinES (50.1K) *ParTUT* (43.5K), *Hindi_HDTB* (281K)
Italian	PoSTWITA	104K	12.8K	13.2K	*ISDT* (294K), ParTUT (52.4K), VIT (241K)

Table 1: Treebanks and number of tokens in train, dev, and test data sets for the target treebanks. Top of table is spoken data, and bottom is for Twitter data. Additional data lists treebanks used for each target treebank, which is in-language unless otherwise noted, and the number of tokens in the training set for each treebank. Treebanks in italics are used in the contrastive data sets.

between the treebanks used, especially for the rather unusual features of spoken data and Twitter. For instance, see Liu et al. (2018) for a discussion of differences between English and Italian Twitter treebanks, or the Naija-NSC documentation for known deviations from UD standards.[3]

To be able to compare the effect of adding in-domain data, we create a contrastive treebank for each IND language of the same size, counted in the number of tokens. We use data from the treebank(s) marked with italics in Table 1.

We think the language sample is interesting and covers many aspects. Even though the majority of languages are Indoeuropean, they mostly have different genera. They range from having hardly any resources like Komi Zyrian, to large resources, like English, and cover some interesting special cases, such as code switching, a Creole language, Naija, and a language with two written varieties, Norwegian.

Parser We use uuparser[4] (de Lhoneux et al., 2017b) which is a transition-based dependency parser using the arc-hybrid transition system with the addition of a swap transition and a static-dynamic oracle, to be able to handled non-projectivity. The parser uses a two-layer BiLSTM as a feature extractor followed by a multi-layer perceptron predicting transitions, in the style of Kiperwasser and Goldberg (2016). Each word, w_i, is represented by the concatenation of a word embedding, $e_w(w_i)$, a character-level embedding, obtained by running a BiLSTM over the characters ch_j ($1 \leq j \leq m$) of w_i, where m is the word length in characters, and a treebank embedding, $e_{tb}(t^*)$:

$$e_i = [e_w(w_i); \text{BiLSTM}(ch_{1:m}); e_{tb}(t^*)] \quad (1)$$

The treebank embedding represents a treebank, t^*, which is chosen among the set of k treebanks used when training the model. During training, t^* is chosen as the treebank to which the current word/sentence belongs. When applying the model, the treebank of the sentence can be used only if the test sentence comes from a treebank that was used during training. In other cases some other method has to be used. In this work we explore the following methods:

- Proxy treebank: when dev data is available, we can try all possible proxy treebanks i.e. all treebanks used during training the model, and choose the treebank, t^*, which performs best on dev data.
- Interpolation: We interpolate the embeddings from all treebanks used during training by averaging them with equal weights: ($t^* = \sum_{t=1}^{k} \frac{1}{k} e_{tb}(t)$)
- Ensemble: We run the model with each possible proxy treebank, obtaining k output trees. Then we apply the reparsing technique by Sagae and Lavie (2006) which applies the Chu-Liu-Edmonds (Edmonds, 1967) algorithm with each arc being weighted by the number of trees for which that arc was predicted.[5]

Note that in all cases we only apply these techniques at test time. The interpolation method only requires a single test run. Proxy treebank requires k dev test runs, followed by a single test run. Ensembling

[3] `https://github.com/UniversalDependencies/UD_Naija-NSC/blob/master/README.md`

[4] `https://github.com/UppsalaNLP/uuparser`

[5] Weighting the arcs by development UAS or LAS instead had little impact on the results, but requires development data.

	Same language		Other language		Spoken						Twitter						**Mean**	
					French		Norwegian		Slovenian		Italian		English		Hindi–English			
	IND	OOD	IND	OOD	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
1	–	–	–	X	18.2	2.8	24.2	8.8	25.9	7.3	27.8	7.2	50.7	33.2	31.4	19.4	29.7	13.1
2	–	–	X	–	21.8	2.2	19.1	3.6	20.7	4.7	23.7	7.5	53.8	40.5	35.0	23.0	29.0	13.6
3	–	X	–	–	74.8	63.4	60.1	52.8	60.2	46.9	71.7	62.8	68.2	55.7	37.0	25.0	62.0	51.1
4	–	X	–	X	75.3	64.3	62.2	54.4	60.1	47.6	72.5	63.4	67.0	54.6	35.9	24.9	62.2	51.5
5	–	X	X	–	75.9	64.5	59.1	52.0	63.2	52.7	73.9	65.5	69.5	58.9	38.0	25.7	63.3	53.2
6	X	–	–	–	76.6	69.4	74.3	69.4	65.8	57.9	82.3	76.9	74.7	68.6	65.0	52.7	73.1	65.8
7	X	–	X	–	76.1	68.8	73.9	67.8	65.3	57.9	81.8	76.3	76.3	70.6	64.1	52.6	72.9	65.5
8	X	X	–	–	84.0	79.2	78.3	73.5	71.8	65.6	84.2	79.4	82.8	78.0	67.6	52.7	78.1	72.1
9	X	X	X	–	83.7	78.0	78.7	73.7	72.7	66.1	84.5	79.5	82.1	77.4	67.2	56.9	78.2	72.0

Table 2: Test set scores for spoken data with different combinations of training data, using the best proxy treebank. For each line, only data sources marked 'X' are used, sources marked '-' are not used. Note that 'Same language' also includes related Slavic languages for Slovenian.

is heavier, requiring k test runs, followed by an application of the CLU algorithm. Interpolation and ensembling both have the advantage of being parameter free, while proxy treebank requires dev data. For languages without dev data we also compare our results to the oracle score, where we pick the best proxy treebank based on test performance.

We use the default hyperparameters of uuparser, as specified in Smith et al. (2018a). Note that no POS-tags are used, since POS-tagging in these difficult domains would lead to the same issues as for parsing. In addition, character embeddings compensate for the lack of POS-tags to a large extent across several typologically different languages (Smith et al., 2018b), and in order for universal POS-tags, the most feasible choice cross-lingually, to be useful for parsing, the tagging quality has to be prohibitively high (Gómez-Rodríguez, 2020). The parser is trained end-to-end on treebank data, without any pre-training. All embeddings are initialized randomly at training time. Each model is trained for 30 epochs, and the best epoch is chosen based on average development scores among treebanks used at training time.

Evaluation Metrics We use unlabelled and labelled attachment score, UAS and LAS, as evaluation metrics. Our system was optimized based on development UAS scores, since we believe that it is a good fit to the case of inconsistent labeling in the treebanks for each target domain. Overall, the test results reflect the trends seen in development data relatively well.

3 Results

We first present results using different sources of training data, IND or OOD, from the same or another language, choosing the best proxy treebank based on development UAS scores. We use the full set of treebanks from Table 1.[6] For other language OOD data, we use the contrastive datasets sampled from the same languages as the other language IND data.

Our main interest is the middle part of Table 2, lines 3–5, where we investigate the effect of adding IND data from other languages to in-language OOD data. Adding out-of-language IND data leads to average improvements of 2.1 LAS points and 1.3 UAS points. It always helps for Twitter, and helps in all cases except Norwegian for spoken data. If we instead add an equivalent amount of out-of-language OOD data, we see minor average gains and a performance that is considerably worse than for IND data. Norwegian is an outlier here as well, with good results for OOD data. We leave an investigation of why to future work. These results confirm that our treebank combination strategy is useful.

The two top lines of Table 2 simulates results when no in-language data is available. As expected these scores are considerably lower than when using in-language OOD data, being so poor that these parsers are hardly useful, confirming previous research, e.g. Meechan-Maddon and Nivre (2019) and Vania et al. (2019). In this case there is no clear difference between IND and OOD data. The scores for English and Hindi–English with IND data are closer to in-language OOD scores, which can be explained by the partial language match between these two treebanks.

As a point of comparison, the bottom part of Table 2 shows the results when data matching both language and domain is available. As expected, it leads to large gains. For all languages, the model trained

[6]Using a subset of these treebanks mostly gave lower scores but showed the same trends.

	Proxy Language			Proxy Domain			Interpolation		Ensemble	
	UAS	LAS	Proxy	UAS	LAS	Proxy	UAS	LAS	UAS	LAS
French	75.9	64.5	fr_partut	76.0	61.7	no_nynorsklia	75.2	63.7	75.6	63.8
Norwegian	59.1	52.0	no_bokmaal	60.2	51.0	sl_sst	61.2	53.5	60.3	50.8
Slovenian	60.7	50.0	sl_ssj	59.6	47.1	no_nynorsklia	60.6	48.8	60.8	47.8
Slovenian+Slavic	63.2	49.6	sr_set	61.3	48.1	no_nynorsklia	63.8	52.2	63.9	48.8
Italian	73.9	65.5	it_partut	67.2	55.2	en_tweet	73.9	64.6	74.4	62.7
English	69.5	58.9	en_partut	66.3	53.8	it_postwita	70.2	61.5	69.4	58.6
Hindi–English	37.6	25.7	en_partut	38.0	26.3	en_tweet	35.4	26.0	37.6	25.6
Mean:	62.8	52.3		61.2	49.0		62.9	52.9	63.1	51.1

Table 3: Test scores for models trained on all available in-language OOD data and IND data from the other languages, using different methods for applying it to the target treebank.

on only the relatively small in-language IND data beats all models trained without it, even though the gap is quite small for French and Slovenian. The gains are especially pronounced for the code-switched Hindi–English and for Norwegian. When in-language IND data is available we see no average gains from adding out-of-language IND data, whereas adding in-language OOD data always helps considerably. We also note that the gap between UAS and LAS gets smaller, when the training data fits the test data better, supporting our intuition that out-of-language OOD data helps more with structure than labels.

Next, we focus on our main scenario of interest, where we have in-language OOD data and out-of-language IND data. We use the model from line 5 in Table 2 and also show results for Slovenian without the additional Slavic languages. We investigate how best to apply the model at test time for cases where the treebank, i.e. the combination of language and domain, has not been seen at training time. We compare using a proxy treebank, matching either language or domain, interpolation, and ensembling. Table 3 summarizes the results.

When choosing a single proxy it is on average 3.3 LAS points and 1.6 UAS points better to use the same language than the same domain, but there is some variation between languages. The interpolation method works well on both metrics, giving the best average LAS scores and competitive UAS scores. Ensembling gives the highest UAS scores by a small margin, but does worse on LAS. We also note that including the related Slavic languages improves parsing for Slovenian considerably, with an LAS gain of 3.4 for the interpolation strategy.

Table 3 also shows the best proxy used, either matching domain or language. For language proxies we note some surprises, Norwegian Bokmaal is a better fit than the matching language variety Nynorsk, and the Serbian corpus is better than Slovenian in the Slavic setting. We also note that the ParTUT treebank is often a good proxy. The differences between proxies are typically small, though. The domain proxies seem more straight-forward, with Norwegian and English being preferred more than the other options. The only small surprise is that Italian was a better fit for English than the partially matching Hindi–English treebank. There could, however, be many reasons for this, such as more similar annotation schemes for Italian and English, or the fact that while there is a partial overlap with English, Hindi is less related to English than Italian.

Finally we apply our methods to the two low-resource languages without any in-language training data. Here, we have no development data for choosing a proxy language, so the focus is on our two parameter free methods: interpolation and ensembling. As a point of comparison we give the oracle score of the proxy treebank with the highest UAS score. We compare three models: using only the close OOD languages from Table 1, and adding either all three IND spoken treebanks or the contrastive OOD treebanks. Results are shown in Table 4. Interestingly, adding the small data from the unrelated languages helps somewhat regardless of if this data is OOD or IND. Adding the IND data do present the overall best scores, though, with the highest UAS scores for Komi Zyrian and the highest LAS scores for Naija. For our target model, interpolation and ensembling works quite well, often tying with the oracle scores, and typically not falling too much behind the oracle. However, in the setting with only related languages, these two methods falls behind the oracle, indicating that these methods works better with a more diverse mix of training languages and domains.[7]

Our experiments confirm the usefulness of our proposed method of mixing training treebanks and

[7]We saw the same trend when we applied these methods to the languages in Table 2.

		Related OOD			Related OOD + other OOD			Related OOD + other IND		
		Oracle	Interp	Ens	Oracle	Interp	Ens	Oracle	Interp	Ens
Komi Zyrian	UAS	32.1	26.4	30.8	32.9	32.9	31.9	35.4	34.4	33.9
	LAS	18.3	14.8	18.4	19.0	19.1	18.2	20.0	19.0	18.7
Naija	UAS	43.1	41.4	41.0	44.1	43.4	43.5	43.2	43.1	44.1
	LAS	28.9	28.0	27.4	29.1	28.6	27.8	30.2	30.0	28.3

Table 4: Test set scores for languages without any training data, using different training data combination, with the oracle proxy treebank, interpolation, or ensembling.

applying the model to new data. Treebank embeddings seem to be capable of encoding aspects both of domain and language.[8] Both interpolation and ensembling have the advantage that they do not require any tuning on development data, which choosing a single proxy does. Interpolation has the further advantage that it requires no extra processing, and seems preferable since it gives the best LAS scores, as well as competitive UAS scores.

4 Conclusion

In this paper we have shown how we can improve parsing for specific domains by combining data in that domain but from another language with in-language out-of-domain data. We show that it is possible to do so using a parsing model with treebank embeddings. We also propose the use of two parameter free methods for applying treebank embeddings to new data at test time, which give competitive results compared to optimizing a proxy treebank based on development data. This indicates that treebank embeddings are able to capture aspects both about text type and language. We also think it is worth noting that in contrast to much previous work, e.g. Smith et al. (2018a), we see gains for languages which are not closely related.

In future work we want to apply our methods also to other text types and to explore how the data selection strategies work with other parsing frameworks. We also want to extend the work on weighted interpolation by Wagner et al. (2020) to the cross-lingual case, to be able to combine it with the proposed methods. Another line of work is to investigate how much annotated data is needed in order to see gains of the same size as when adding IND treebanks from other languages.

In this work we did not take advantage of any type of pre-trained word embeddings. It is likely that either cross-lingual static word embeddings (Ruder et al., 2019) or multilingual dynamic word embeddings, like multilingual BERT (Devlin et al., 2019) could improve the results overall. Using either of these resources would also allow us to utilize IND in-language unlabeled data in the pre-training step, which might potentially lead to improvements. We do believe that seeing labelled data, with arc types that are specific to the text types in question, as we do in this work, is also useful. It is an open question, which we leave to future work, how pre-training would interact with our proposed method.

Acknowledgments

Thank you to current and former members of the Uppsala parsing group for many fruitful discussions: Ali Basirat, Daniel Dakota, Miryam de Lhoneux, Artur Kulmizev, Joakim Nivre, and Aaron Smith. I would also like to thank the anonymous reviewers for their insightful comments.

References

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 2440–2452, Minneapolis, Minnesota, US.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016a. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

[8] We also experimented with separate embeddings for domain and language, which gave lower scores.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016b. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017a. 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.

Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2017b. Arc-hybrid non-projective dependency parsing with a static-dynamic oracle. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 99–104, Pisa, Italy.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 4171–4186, Minneapolis, Minnesota, US.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240.

Mark Anderson Carlos Gómez-Rodríguez. 2020. On the frailty of universal POS tags for neural UD parsers. In *Accepted to CoNLL 2020*.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. Frustratingly easy neural domain adaptation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 387–396, Osaka, Japan.

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.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 2779–2795, Hong Kong, China.

David Y. W. Lee. 2001. Genres, registers, text types, domains, and styles: Clarifying the concepts and navigating a path through the BNC jungle. *Language Learning & Technology*, 5(3):37–72.

KyungTae Lim, Cheoneum Park, Changki Lee, and Thierry Poibeau. 2018. SEx BiST: A multi-source trainable parser with deep contextualized lexical representations. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 143–152, Brussels, Belgium.

Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A. Smith. 2018. Parsing tweets into universal dependencies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 965–975, New Orleans, Louisiana.

Ailsa Meechan-Maddon and Joakim Nivre. 2019. How to parse low-resource languages: Cross-lingual parsing, target language annotation, or both? In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 112–120, Paris, France.

Joakim Nivre, Mitchell Abrams, Željko Agić, et al. 2019. Universal dependencies 2.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, 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 2020)*, pages 4034–4043, Marseille, France.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. Towards a continuous modeling of natural language domains. In *Proceedings of the Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods*, pages 53–57, Austin, TX.

Sebastian Ruder, Ivan Vulić, and Anders Sogaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:69–631.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 129–132, New York City, USA.

Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. 2017. Adversarial training for cross-domain universal dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 71–79, Vancouver, Canada.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018a. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium.

Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018b. 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.

Sara Stymne, Miryam de Lhoneux, Aaron Smith, and Joakim Nivre. 2018. Parser training with heterogeneous treebanks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 619–625, Melbourne, Australia.

Marlies van der Wees, Arianna Bisazza, Wouter Weerkamp, and Christof Monz. 2015. What's in a domain? Analyzing genre and topic differences in statistical machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, Short Papers*, pages 560–566, Beijing, China.

Clara Vania, Yova Kementchedjhieva, Anders Sogaard, and Adam Lopez. 2019. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. 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 1105–1116, Hong Kong, China.

Joachim Wagner, James Barry, and Jennifer Foster. 2020. Treebank embedding vectors for out-of-domain dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8812–8818, Online.

Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big Data*, 3(1):1–40.

Qiao Xiuming, Zhang Yue, and Zhao Tiejun. 2019. Learning domain invariant word representations for parsing domain adaptation. In *Natural Language Processing and Chinese Computing (NLPCC 2019)*, pages 801–813, Dunhuang, China.