Resourceful at Any Size: A Predictive Methodology Using Linguistic Corpus Metrics for Multi-Source Training in Neural Dependency Parsing

Ajda Gokcen

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2021

Reading Committee:
Gina-Anne Levow, Chair
Richard Wright
Mari Ostendorf

Program Authorized to Offer Degree:
Linguistics

University of Washington

ABSTRACT

Resourceful at Any Size: A Predictive Methodology Using Linguistic Corpus Metrics for Multi-Source Training in Neural Dependency Parsing

Ajda Gokcen

Chair of the Supervisory Committee:
Gina-Anne Levow
Department of Linguistics

Multilingual modeling comes up in natural language processing at any scale. High-resource language corpora train high-performing models, and can be combined with other language corpora of all sizes to make better models for low-resource languages. Projects like Universal Dependencies even make it possible to train highly multilingual models from standardized morphosyntactic labels. Multilingual (or, more generally, multi-source) training does not consistently improve modeling performance, however. With an abundance of language resources comes a difficult design choice: which corpora will train better together rather than separately? More specifically, when is it worthwhile to supplement (i.e., concatenate) one corpus with another during training, rather than training on the first corpus alone? Approaches to selecting and evaluating candidate combinations tend toward two extremes: ad hoc or exhaustive. In this work, I put forth an alternative, predictive methodology for outcomes of concatenative training in dependency parsing. I leverage treebanks from the Universal Dependencies framework to assess the utility of linguistic corpus metrics in multi-source modeling. This approach is both robust and practical, using computationally simple metrics that expand upon intuitions of linguistic similarity, and making it possible to reasonably predict which conditions will yield significant improvement for a target corpus. Although the results are specific to a particular family of models and the task of dependency parsing, the approach holds promise for any number of natural language processing applications.

Acknowledgments

To Jiahui, for being the best of us.

To my mom, for being the wind beneath my winds and the fire under my butt.

To my dad, for dreaming with me and doing more of the work than he'll take credit for.

To my brother, for being my fellow traveler.

To my sister, for being my partner in crime and carbohydrates.

To Amanda, for making anywhere feel like home.

To Elizabeth, for the first decade's worth of cahoots.

To Biscuit, for being the distinguished feline in all of the example sentences.

To Charlie, for getting there first and lighting the way.

Table of Contents

List of Figures

List of Tables

Part I

Foundation

CHAPTER 1
Introduction

1.1 Motivation

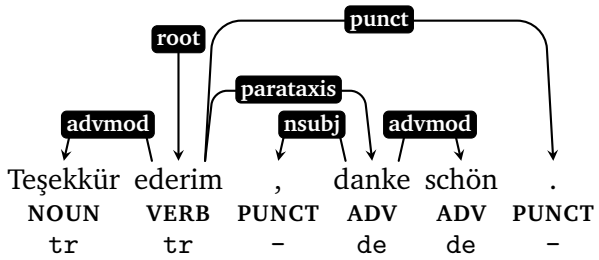
When are additional data useful data?

Even in state-of-the-art *natural language processing* (NLP), which is very much a part of the big data zeitgeist, it's an oversimplification to say *more* is inherently *better* when it comes to training data. Supplementing a language corpus with numeric meteorological data, as an extreme example, would be unlikely to add any benefit to training a model for an NLP task beyond adding noise. Supplementing that same language corpus with a similar language corpus – such as web-scraped blog post data in addition to news publication data – probably *would* improve modeling.

Most cases will fall somewhere in-between. On the one hand, a model for an NLP task trained on text from two very different languages, like English and Japanese, might at best perform as well as two separate models trained on the language corpora independently. On the other hand, corpora featuring languages closely related by descent or by contact might add up to more than the sum of their parts. That is, a model trained under such multilingual conditions might perform better on at least one of the corpora than would a monolingually-trained model.

Perhaps the clearest case for this is *code-switching*, where a single conversation or even a single utterance alternates between two or more languages, as in this dependency-parsed Turkish-German sentence:

(1)

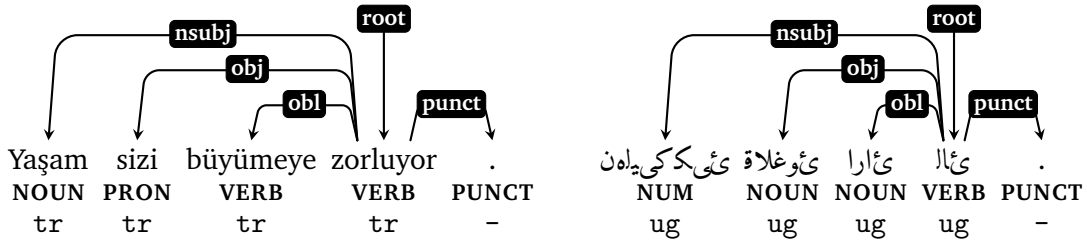


(Çetinoğlu and Çöltekin, 2019)

In Example (1), a Turkish *thank you* is followed by a German *thank you*. Corpora for code-switching, which are effectively multilingual on their own, are fairly rare; this example comes from a small corpus transcribed from spoken data. Bringing in purely German and/or Turkish corpora to supplement the code-switching data is a clear step toward a more robust Turkish-German model.

Such an example, with whole words and phrases in common across languages, is likely to benefit from additional language corpora without having to bring in any question of structure. In other cases, there might be structural similarities between two very different-looking languages that a syntax-aware model could take advantange of. Take these Turkish and Uyghur sentences:

(2)



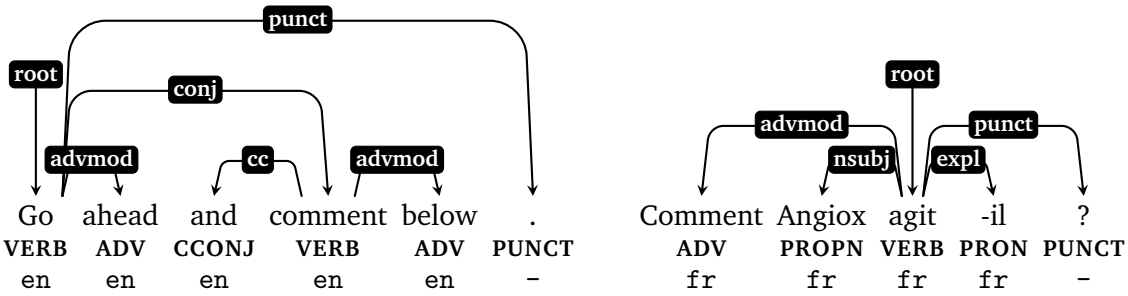
(Türk et al., 2020) (Aili et al., 2016)

Both sentences in Example 2 have the same dependency structure – and, at the same time, completely different writing systems: Turkish uses a Latin script, and Uyghur an Arabic one. The syntactic similarity is high because both languages are in the Turkic family; the orthographic similarity is low because of a fairly complex series of historical events, including a switch from Arabic to Latin script in Turkey nearly a hundred years ago. As interesting as the history may be, the important points are all in the data: similar syntactic structures, disjoint character sets.

Of all the Turkic languages, Turkish is the only one that approaches the high-resource level, and Uyghur is not the only other Turkic language with a non-Latin script. This could spell trouble for trying to use Turkish to supplement other Turkic language corpora in any *lexicalized* model (i.e., one that uses word-specific information). Since the languages share no (non-punctuation) characters, they certainly share no exact words or phrases, which makes the case for supplementing Uyghur with Turkish data much weaker than the case for supplementing Turkish-German code-switching.

Other language combinations might face the opposite scenario, – where corpora appear to share words that actually have very different meanings – as in these English and French sentences:

(3)



(Zeldes, 2017) (Candito and Seddah, 2012)

A word spelled as *comment* appears in both sentences in Example 3, but one is a verb and the other an adverb (meaning *how*). English and French are related as two Indo-European languages, but many commonalities come from more recent borrowing from French into English. With full words matching in form while behaving differently syntactically, it's not clear whether combining these two (high-resource) languages would be helpful, or whether it might even be a hindrance.

All of these examples show the breadth of ways in which different language corpora might interact with one another based on various dimensions of similarity. Prior linguistic knowledge goes a long way for predicting such interactions and thus, for example, how successful a dependency parser might be when trained on both corpora concatenatively. That knowledge is not enough on its own, however, for successful selection of *multi-source training conditions* – that is, where each corpus is a source, the goal is to select the set of sources (of the same or of different languages) that, when concatenated, will train a model with optimal performance on a target test corpus.

Several factors make prior linguistic knowledge insufficient as a strategy on its own. First, it's impossible to know every interaction between all natural languages *a priori*. Corpus combination is, by definition, a combinatorial problem; even sticking to corpus *pairs*, the number of possible interactions to track explodes quickly when enough different language corpora are available.

Second, NLP models do not deal with languages as complete, theoretical entities; they deal with corpora, which represent only a sample of a natural language. At best, these samples are representative, but still incomplete. Different corpora of the same language behave differently – even those in the same domain/genre. An English news corpus from 2020 would include articles about different topics (and thus include different words) compared to one from 2010.

Finally, NLP models are beholden to their input representations. Turkish and Uyghur might train a better parser concatenatively if the input used *delexicalized* (i.e., with only *parts-of-speech*, or POS) rather than lexicalized representations – let alone character-aware ones. Leaving out the lexical information for Turkish-German, meanwhile, would omit a lot of information vital to parsing performance. Additionally, learning word embeddings that ignore characters might work well enough, but a purely word-level (morphologically naïve) model for a morphologically complex language like Turkish is weaker than nearly any character-aware model. All of which is to say: the actual effect of the interaction between languages is a function of both the corpora *and* the particular model.

While prior knowledge of cross-linguistic similarities may not be enough to inform perfect choices for multi-source training conditions, it can point in the right direction. Exhaustive training and testing of models on all possible corpus pairs – or larger combinations – is not at all feasible if more than a handful of corpora are available. Applying prior knowledge to prune the search space of unlikely combinations, namely pairs of corpora for languages that are unrelated and/or completely dissimilar, is a good enough strategy for most applications, if theoretically unsatisfying.

I propose an empirically-grounded alternative: a *predictive* methodology for the assessment and selection of multi-source training conditions using corpus metrics. Compared to the exhaustive approach, combinatorial search that involves fairly simple calculations of corpus metrics, as opposed to the training of as many neural models as there are unique corpus pairs, is a much more practical solution – assuming the calculations can be translated into reasonable predictions.

Prior linguistic knowledge pertinent to a corpus ought to be measurably reflected in the data, after all. The question of *how similar is similar enough?* for training corpora together is reminiscent of questions in theoretical and corpus linguistics. Is it possible, for example, to quantify mutual intelligibility between two related languages? Cross-linguistic corpus metrics, based in calculations such as entropy between corpora, correlate with how well speakers of different languages can understand one another. It seems likely that they might also correlate with how well one language corpus does at improving model performance for another corpus.

1.2 Problem Definition & Approach

To address the question of how to determine when concatenative training will prove beneficial over independent training, I define the problem as having the following components:

- a **set of natural language text corpora**, U, each annotated with a single set of labels;
- a **target corpus**, $t \in U$, which is split into a training set t_r and test set t_e;
- a **set of supplemental corpora**, $S \subseteq U$, where each corpus $s_i \in S$ represents a training set;
- a **model architecture**, M, that can then be trained on the corpora in U (either individually or concatenated in pairs or larger combinations) and tested on the corresponding test sets, where the output of a model trained on t_r and tested on t_e is denoted $M(t_r, t_e)$; and
- an **evaluation metric**, E, that can be used to sort model outputs by performance.

Given these components, I want to be able to determine which elements of S make it true that $E(M((t_r \cup s_i), t_e)) > E(M(t_r, t_e))$ – assuming E is a maximizing function. As a corollary, it should also be possible to determine the s_i with the maximum $E(M((t_r \cup s_i), t_e))$.

In other words, given a target corpus and a set of candidate supplemental corpora, the goal is to find the corpora[1] that, when concatenated with the target training set, can train a model *(multi-source)* with performance on the target test set that is significantly better than that of a model trained on the target training set alone *(single-source)* – or to determine that none of the candidate corpora can do so, and thus single-source training is preferable. Put even more simply: I want to select the best set of corpora *(conditions)* to train a model for a target corpus.

More specifically, I want to select effective supplemental corpora from among the candidates *without* needing to train and evaluate models for every possible multi-source training condition. Instead, there ought be some function F for which $F(t, s_i)$ approximates the same truth value as $E(M((t_r \cup s_i), t_e)) > E(M(t_r, t_e))$ – i.e., F uses some corpus-based, model-independent[2] metrics to predict the relative efficacy of each potential multi-source model over a single-source model. This approximation can then be used to select the conditions worth training.

[1] While multi-source conditions *can* involve arbitrarily-many corpora, I focus on cases with 1-2 supplemental corpora.

[2] *Model-independent* meaning that metrics are derived from corpora alone, even if *conceptually* tailored to the models.

This constitutes a specific type of transfer learning and data selection. The task and type of model are fixed, and all multi-source conditions are concatenative – assuming the presence of some amount of target training data, however sparse (i.e., *multi-source* includes the target corpus as one of the sources). There is also the built-in assumption that the training and test portions of the target corpus are split randomly enough to at least approximate two independent samples of the same "population" of data – that is, they should have similar distributions by all measures.

As for the testbed I use to go about the problem, I start from a fixed foundation of (a) neural shift-reduce dependency parsers that are (b) trained on *Universal Dependencies* (UD) corpora with non-empty training sets – including the sources for all prior example parses – and (c) evaluated in terms of *labeled attachment score* (LAS). I focus on conditions without additional unlabeled training data, as these may be unavailable for low-resource languages.

Beyond the set of single- and multi-source conditions used, the main point of exploration is in the metrics. My goal is to explore an array of metrics that capture both data sparsity (i.e., the strength of the training set, in quantity and quality) and corpus similarity and to use basic computational tools – like regression and threshold-finding – to turn these into predictions for both single-source performance and multi-source improvement.

The use of only the one type of model and task (and lack of pretraining, standard in modern NLP) limits direct applicability. My goal, however, is not to generate a fixed set of metrics with exact numbers and functions to be applied wholesale to other NLP applications. Rather, the generally applicable elements should be the family of metrics used to capture various dimensions of linguistic corpus similarity as well as the ethos of choosing these metrics based on the models and task.

I hypothesize that, in capturing intrinstic corpus sparsity, text- and label-based facets of cross-corpus similarity, and asymmetric cross-corpus compatibility, I can approximate the relative performance of multi-source training conditions with greater efficiency than training all the models in question. Furthermore, I hypothesize that these metrics can capture linguistic similarities consistent with external linguistic knowledge, such as language relatedness. I find that said metrics are indeed strong predictors with minimal adjustment, and that they hold promise for adaptation to other NLP applications and for bringing the strengths of corpus linguistics to NLP in general.

1.3 Contribution

In this dissertation, I aim to create a process that, given a target corpus and set of candidate supplemental corpora, reliably predicts whether it's worthwhile to try training models in concatenative, multi-source conditions – and, if so, which supplemental corpora are the most promising. To this end, I build out and test a metric-focused methodology for a specific family of lexicalized neural dependency parsers, with the goal of outlining an example for applying similar-but-adapted methodologies to other models and tasks. The steps toward developing this methodology include:

- putting forth a suite of **intra-corpus metrics** to find which correlate best with single-source model performance (and therefore sparsity);
- putting forth a suite of **cross-corpus metrics** to find which best differentiate between multi-source models that significantly improve on a single-source baseline and those that don't;
- training a suite of **neural dependency parsers** under single- and multi-source conditions for testing out the predictive power of the metrics;
- using the most robust metrics, correlations, and thresholds to construct a process for predicting **significant multi-source improvement**;
- analyzing **corpora of interest**, including but not limited to code-switching and low-resource languages, to see where the predictive process would have succeeded or failed and why; and
- discussing which aspects of the process are generalizable for **other NLP applications**.

I find that some of the metrics tested have very strong predictive power: namely, metrics that account for the similarity between the target and supplemental corpora *and* the intrinsic sparsity of the target corpus. In other words, the smaller the training portion of the target corpus and the more similar the supplemental corpus, the more effective multi-source training will be.

The robustness of the predictive methodology is both theoretically satisfying – given that the metrics largely reflect lingustic intuitions in a quantifiable way – and promising for the possibility of adaptation to other models and corpora. Linguistic corpus metrics have real value in assessing NLP corpora, and there is much to be gained from incorporating them into the model-making process.

1.4 Outline

The remainder of this dissertation is divided into eight chapters, as follows.

Chapter 2 details related work in both linguistic corpus metrics and dependency parsing, including some works that also construct predictive approaches to multilingual model training.

In Chapter 3, I go into depth about my approach to the task of dependency parsing (specifically shift-reduce dependency parsing) and give further details regarding the UD framework and the array of UD corpora used to train and test models.

Chapter 4 pertains to the architecture and training strategies for the family of neural dependency parsers I use as the models for all single-source baselines and multi-source experiments.

In Chapter 5, I outline the aims and bases of the metrics used to predict both single-source performance and multi-source improvement, in addition to demonstrating some of their basic utility in capturing facets of individual corpora and similarity between corpora.

In Chapter 6, I present the results of the single-source models first in terms of basic parsing performance and then in terms of the metrics. The latter portion involves running a suite of tests to find the metrics that best predict performance, so that the select metrics can then be used in the multi-source experiments.

Chapter 7 contains the results of the multi-source models, with a focus on the parsing performance relative to the single-source baselines. I run another suite of tests to find the metrics with thresholds that best recreate the threshold of significant improvement over the baselines, and propose using these as a predictive methodology. I then look at case studies within the data to see how well the predictions capture the observed patterns among the corpora.

Finally, Chapter 8 is where I discuss the overall findings and implications of the work before concluding in Chapter 9.

CHAPTER 2
Related Work

2.1 Linguistic Corpus Metrics

2.1.1 Metrics for Linguistic Study

Linguistic corpus metrics, in general, are tried and tested with the goal of quantifying known linguistic phenomena and relationships. Some of the inspiration for the metrics in the current work comes from cross-linguistic study separate from NLP.

In particular, Moberg et al. (2007) and Frinsel et al. (2015) quantify *asymmetric intelligibility* among Scandinavian languages. That is, Scandinavian languages are similar enough that speakers can communicate with one another easily in their own respective languages. The exact degree of understanding varies not only by language pair, but also by direction; Danish speakers understand Swedish more readily than Swedish speakers understand Danish, for example.

The aforementioned works specifically use *conditional entropy* to quantify this phenomenon, finding that the resultant calculations on spoken and written language also manage to correctly reflect greater asymmetry in the case of spoken language than in the case of written language. Kyjánek and Haviger (2019) find the metric highly effective in reflecting asymmetric intelligibility among West Slavic languages, as well.

This kind of directional metric between language pairs is relevant to multi-source training because, in the same way that understanding Danish helps with understanding Swedish more than vice versa, a large corpus will probably improve parsing results for a small corpus of a similar language more than vice versa. The same pattern could extend beyond corpus size, as well; while I don't employ conditional entropy in particular (in part because it involves tracking *cognates*, or words known to share a common ancestor), corpus metrics for predicting the efficacy of concatenative training should also include the possibility of asymmetry when the roles of target and supplemental corpus are reversed.

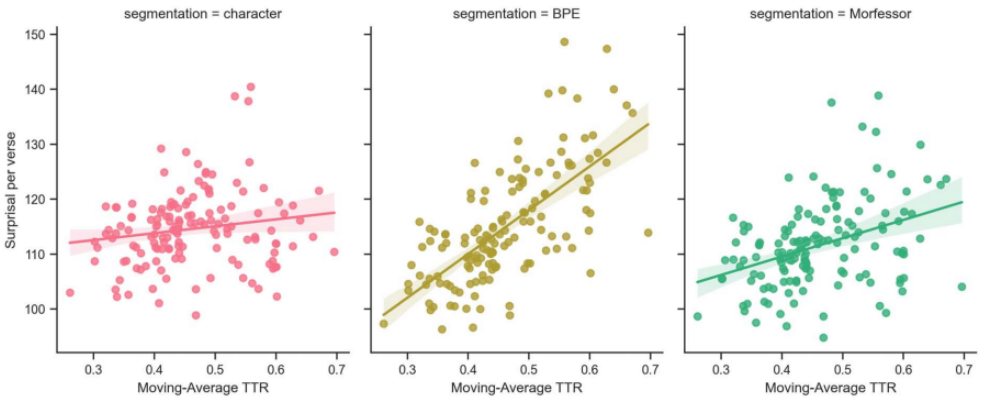


Figure 2.1: A graph from Park et al. (2021) showing the interaction between the type of model and the difficulty that family of models has with languages across the scale of morphological complexity.

2.1.2 Metrics for NLP

Other relevant works utilize corpus metrics for analyzing or improving NLP. Several works have aimed to capture morphological complexity and its impact on language modeling difficulty, with mixed findings. Cotterell et al. (2018) use lexicon-based *counting complexity* of morphological categories to compare languages by performance in character-level language modeling, taking care to control for effects of length and spelling. While they initially find that morphologically complex languages are harder to model, Mielke et al. (2019) continue the research and find that the association disappears with different models, leading to a reversal of the conclusion.

Park et al. (2021), however, use more corpus-based metrics, concluding that the association between complexity and difficulty indeed exists, even if it's not entirely straightforward. They focus on metrics like *type-token ratio* (TTR), namely the *moving average* (MATTR) variant that controls for the effect of corpus size. Figure 2.1 shows how their experiments with different segmentation schemes, including both character-level models and models using larger subword units, have varying degrees of sensitivity to morphological complexity, as measured by MATTR.

I use similar ideas in the current work, – namely, TTR to capture complexity, morphological and otherwise – although I avoid controlling for size effects. Some metrics are effective because they capture the fact that both corpus size *and* complexity are factors in model performance – and that, therefore, the more complex languages need more data to perform as well as their counterparts.

Ruder and Plank (2017) use metrics to train models that learn to select data for transfer learning across several testbed tasks, including dependency parsing. Unlike the current work, they are not focused on multilingual transfer – they instead look at transfer across domains, models, and tasks. I use many of the same basic metrics and components for *similarity* and *diversity*: cosine similarity, vocabulary/term distributions, TTR, and entropies. Ruder and Plank (2017) learn optimal linear combinations of metrics, using Bayesian Optimization, to select samples to train from for a given domain or task. This also differs from the current work in that it's selecting individual samples rather than taking a corpus as a whole.

2.1.3 Typological vs. Corpus Measures

As corpus metrics are developed to quantify existing linguistic knowledge, the question of how well the metrics capture *typological* information (i.e., expert labels of the language as a whole, not specific to a corpus) can evolve a step further. Can corpus metrics capture additional complexities that typological data cannot – or has not?

Berdicevskis and Bentz's (2018) Shared Task on Measuring Language Complexity includes seven entries dedicated to linguistic complexity measures derived from UD treebanks. Çöltekin and Rama (2018) put forth several text- and label-based metrics of morphosyntactic complexity, including TTR over a fixed window and POS bigram perplexity. von Prince and Demberg (2018), in another entry, explore POS trigram perplexity for measuring syntactic complexity – predicting (and confirming) that languages like Turkish should get a high perplexity score that reflects flexible word order. Measures of complexity based on label n-grams like this, both entropy-related and otherwise, are a major contingent of the metrics used in the current work.

The entries in the shared task, as a whole, find that corpus-based metrics are either consistent with or stronger than typological designations. This finding is much like one by Bentz et al. (2016) – namely, that language-wise corpus metrics of complexity (like entropy and TTR) correlate strongly with metrics derived from WALS features (see Figure 2.2). Although the only expert labels in the current work are of language relatedness, I find that corpus similarity metrics are reliably higher for corpora of more closely related languages, among other patterns consistent with linguistic intuition.

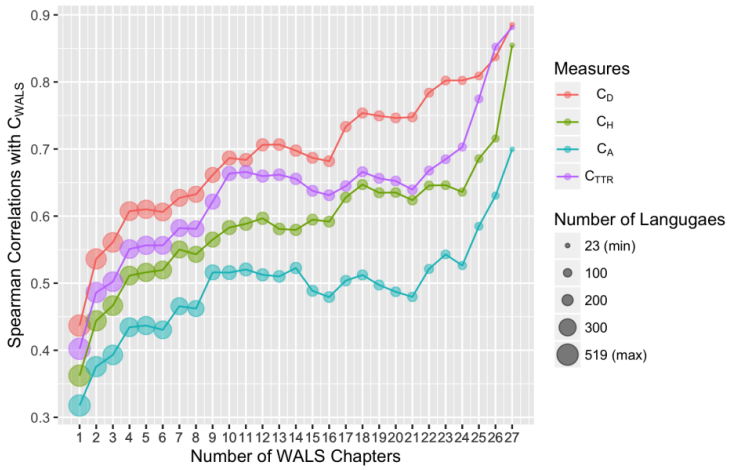


Figure 2.2: A graph from Bentz et al. (2016) showing the correlation of corpus-based metrics of complexity to a WALS-derived metric. Not all languages have values for all WALS features; the correlations are stronger for languages that have values for more of the features.

2.2 Multilingual Dependency Parsing

2.2.1 Broadly Multilingual Models

Some multilingual parsing studies have a fixed set of training languages for all models. Ammar et al. (2016) use typological properties of languages, taken from WALS online (Haspelmath et al., 2005; Dryer and Haspelmath, 2013), to train multilingual dependency parsers for an array of Germanic and Romance languages. The typological features help to mitigate the effects of contradictory patterns across languages as much as to learn from cross-linguistic similarities. Naseem et al. (2012) also train a WALS-informed multilingual dependency parser, specifically to transfer the performance of multiple high-resource languages to zero-resource languages.

Duong et al. (2015) find an interesting pattern in the utility of (delexicalized) multilingual training, as shown in Figure 2.3. A multilingual model that hasn't been trained on data from the target language is better than a model that *has* been trained on data from that target language when the training corpus is small enough. This improvement diminishes as the training corpus gets larger, until disappearing entirely once the corpus reaches 20K tokens or so. This turns out to be consistent with findings in the current work comparing multi-source to single-source conditions.

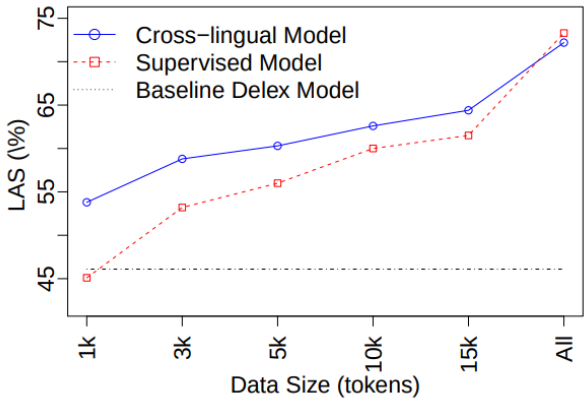


Figure 2.3: A graph from Duong et al. (2015) demonstrating that, the smaller the training data, the more improvement multilingual (sans target language) models provide over supervised models.

Some of the earlier work in cross-linguistic transfer for parsing includes that of McDonald et al. (2011), who find success in delexicalized multi-source transfer for dependency parsing of languages with no resources (i.e., zero-shot learning). Prior to that, Zeman and Resnik (2008) present possibly the earliest attempt at multi-source parsing for related languages (specifically Swedish and Danish).

2.2.2 Selectively Multilingual Models

More recently, the CoNLL 2018 Shared Task (Zeman et al., 2018) – and the one in 2017 before that (Zeman et al., 2017) – had participants train dependency parsers on UD corpora to be tested on 82 different treebanks. Systems had to take raw text (i.e., characters) as input, and test sets included those with all levels of training data availability – from high-resource to low-resource to zero-resource. As such, many of the submitted systems employed multi-source (including multilingual, but also monolingual, cross-domain) methods for the low-to-zero-resource cases.

Che et al. (2018), in presenting the winning system of the more recent shared task, make extensive use of concatenative training for corpora of the same language, and find great success in doing so. They experiment with similar methods of cross-lingual corpus concatenation, with not nearly as much success. This indicates, they conclude, that "in cross-lingual parsing, sophisticated methods like word embeddings transfer... and treebank transfer... are still necessary."

	ug_udt		uk_iu		ga_idt		sme_giella
ug_udt	**69.27**	uk_iu	88.84	ga_idt	**62.84**	sme_giella	**66.33**
+tr_imst	19.27	+ru_syntagus	**90.74**	+en_ewt	51.00	+fi_ftb	59.86

Figure 2.4: A table from Che et al. (2018) showing the results of experiments with cross-lingual concatenative training on UD corpora for several related languages. Only the combination of Ukrainian and Russian improve results over a monolingual (Ukrainian) baseline.

Figure 2.4 shows that the concatenative training of Turkish and Uyghur – notably, the only of the four corpus pairs where the related languages use different writing systems – is particularly detrimental for their models. Separately, they employ an interesting method of selecting supplemental corpora for low-resource target corpora: they test all of the shared task's provided baseline parsers on a target corpus and pick the language corpus that trained the parser with the best performance.

Two prior works specifically use predictive methodologies of supplemental corpus selection in multilingual dependency parsing: Rosa and Žabokrtskỳ (2015) and Lin et al. (2019). Rosa and Žabokrtskỳ (2015) use a metric in the same vein as Çöltekin and Rama (2018) and von Prince and Demberg (2018). Rather than looking at a single corpus's POS bigram or trigram perplexity, however, they use a metric for cross-corpus comparison: Kullback-Liebler (KL) divergence (Kullback and Leibler, 1951) for the POS trigram distributions of target and transfer corpora.

They use variants of this metric for selecting training corpora in a delexicalized, zero-shot setting, finding that it selects the best language just under half of the time and significantly boosts performance when used for treebank weighting. Related metrics, like cosine similarity of the distributions, are also employed, with less success. In this work, I use very similar metrics – in particular, I use the related metric of cross-entropy to compare label n-gram distributions. Cosine similarity also turns out to yield more success than it seems to for Rosa and Žabokrtskỳ (2015).

Lin et al. (2019) use both corpus-specific and typological measures for corpus selection in a multilingual setting. They ask: *given a particular task low-resource language and NLP task, how can we determine which languages we should be performing transfer from?* The calculated metrics are not just compared but combined into trained models for ranking candidate corpora. A diagram of the workflow is shown in Figure 2.5.

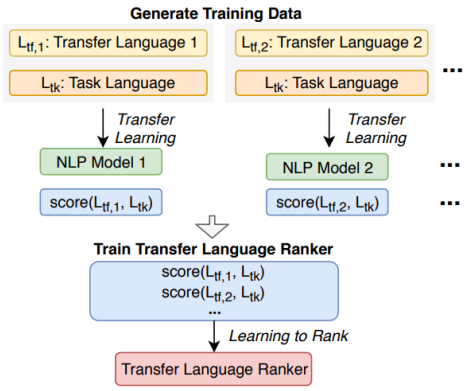


Figure 2.5: Lin et al.'s (2019) workflow for ranking *transfer languages* (referred to in the current work as *supplemental corpora*) in terms of improvement in training *task languages* (referred to in the current work as *target corpora*).

These two works differ from this one in a few key ways. Terminology-wise, I avoid using the general term *transfer* to refer to my models, which are strictly concatenative (i.e., both target and supplemental corpora are used in training). Lin et al. (2019) use *transfer* to refer to both concatenative training and other modes of training – including using pre-trained multilingual embeddings, or training only on the supplemental/transfer corpora (as do Rosa and Žabokrtskỳ, 2015).

Additionally, both Rosa and Žabokrtskỳ (2015) and Lin et al. (2019) work from a strictly *multilingual* rather than possibly-monolingual *multi-source* setting. As such, they can refer to *corpora* as *languages* without losing much clarity. In contrast, the strict *corpus-as-data* vs. *language-as-phenomenon* distinction I try to maintain is partially a consequence of the corpora in use; the UD framework (see Chapter 3; Nivre et al., 2016; de Marneffe et al., 2021; Zeman et al., 2021) has many languages represented by multiple corpora apiece.

I include corpus pairs of the same language as well as those of different languages, in addition to maintaining deference to the complexity of drawing lines between languages. The code-switching Turkish-German corpus (as seen in Example 1) is a prime case. Is it German? Is it Turkish? Is it both, or neither? The answers shouldn't matter too much as long as the implications about linguistic – and therefore corpus – similarity can be quantified. The same metrics that capture similarity between corpora of different languages should be able to capture the degree of similarity between corpora of the same language – or, as it were, corpora of possibly-overlapping languages.

Like Rosa and Žabokrtskỳ (2015) but unlike Lin et al. (2019), I'm not solely interested in low-resource target corpora, although those are certainly the strongest use case for multi-source training. A target and supplemental corpus may both be of the same language, and may both be high-resource. Any two corpora, of whatever size and relation to one another, should be quantifiable by the same metrics, occupying different ends of a scale.

Where Rosa and Žabokrtskỳ (2015) focus only on POS-based features (which is reasonable, given that's the input to their delexicalized models), I also look at features for other labels as well as word- and character-level information. Where Lin et al. (2019) combine corpus-based measures (like size, TTR, and subword-overlap – similar to the word- and character-level features I use) with typological features from other resources, I stick to corpus-based features alone (except for a few language-relatedness baselines). Where Rosa and Žabokrtskỳ (2015) focus on selecting the best transfer corpus, and Lin et al. (2019) use a heavy-weight learning-to-rank approach with their metrics, I look at finding *all* supplemental corpora that yield improvement and use simple correlations and thresholds in order to keep the focus on the metrics themselves.

Basically, my contribution to this thread of work in using predictive metrics for selecting multi-source conditions in dependency parsing is to expand (a) the kinds of corpus pairs under analysis, and (b) the breadth of corpus-based metrics in use.

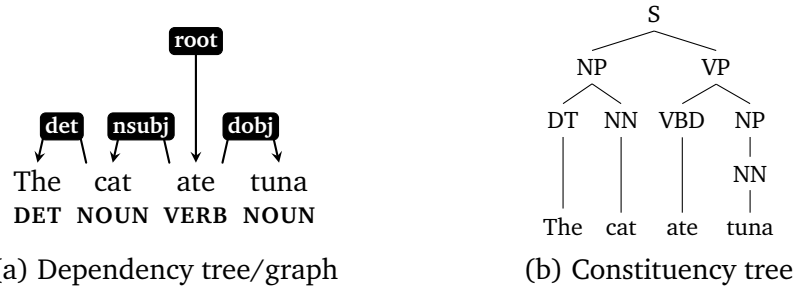
CHAPTER 3
Task & Data

3.1 Transition-Based Dependency Parsing

3.1.1 Dependency Trees

While there any number of theories regarding the syntactic structure of natural language and how to represent it, for the purposes of NLP, there are two overarching categories of representation: *dependency trees* and *constituency trees*. In constituency trees, individual words of an utterance are leaf nodes combined and subsumed under nodes representing phrases, which are themselves recursively combined and subsumed under parent phrasal nodes. In dependency trees, there are only as many nodes as there are words (possibly plus one representing the *root* of a sentence), as relations are directed arcs from *head* to *dependent* words.

Figure 3.1 compares dependency and constituency representations for a simple sentence. The dependency tree is strictly smaller, without any phrases, but the flattened form lacks the expressivity of the constituency tree (e.g., the combination of verb and object into a verb phrase prior to joining with the subject). The relative simplicity of dependency annotations, however, has to some extent made them the default representation for parsing in modern NLP. Certainly some of the largest collections of multilingual corpora for parsing are annotated with dependencies – namely, UD.



(a) Dependency tree/graph (b) Constituency tree

Figure 3.1: Two representations of the syntactic structure for the sentence *The cat ate tuna.*

This abundance of data is one of the primary reasons I've chosen dependency parsing as my testbed task for predictive metrics and multi-source modeling. Dependency annotations are, if not the most complete representations of syntax, certainly expressive enough to provide a foundation for exploring structure in language data – and lending themselves to corpus metrics that reflect typological linguistic features (Bentz et al., 2016; Berdicevskis and Bentz, 2018).

For much of the history of NLP, constituency trees and parsing were the dominant paradigm, though dependency trees and parsing have always existed in parallel. It's possible, if difficult to confirm, that dependencies were more popular in the USSR in the early days of NLP, and might predate constituency trees altogether in one academic tradition or another (Hays, 1958; Kulagina et al., 1958; Owens, 1988).

Dependency parsing started to catch up in popularity once Nivre (2003) and Yamada and Matsumoto (2003) put forth efficient linear-time dependency parsers using a *shift-reduce* algorithm. Although the algorithm had been in use in computing (Knuth, 1965; Korenjak, 1969) and linguistics (Marcus, 1978) for some time, its greedy approach made it less viable than globally optimal parsing algorithms. Prior to this, dependency parsers generally used the same cubic-time chart parsing algorithms common for constituency parsers (Collins, 1996; Eisner, 1996a).

While shift-reduce algorithms are still used for dependency parsers, graph-based parsing algorithms (McDonald et al., 2005) have become at least as common. Despite being theoretically less efficient (i.e., quadratic-time) than the linear-time shift-reduce parsers, which are generally called *transition-based* systems, graph-based systems are globally optimal and lend themselves more directly to modeling dependency graphs that may not be perfectly formed trees – which are not uncommon in natural language.

In any case, I stick to transition-based systems in the current work due both to their efficiency and to a certain aspect that's especially helpful in the derivation of syntactic corpus metrics: the *oracle* transition sequences central to the shift-reduce algorithm.

ROW #	TREE	STACK	BUFFER	NEXT TRANSITION
1	-	[]	[The cat napped .]	$\text{shift}_{\text{DET}}$
2	The **DET**	[The]	[cat napped .]	$\text{shift}_{\text{NOUN}}$
3	The cat **DET NOUN**	[The cat]	[napped .]	left_{det}
4	det The cat **DET NOUN**	[cat]	[napped .]	$\text{shift}_{\text{VERB}}$
5	det The cat napped **DET NOUN VERB**	[cat napped]	[.]	$\text{left}_{\text{nsubj}}$
6	det nsubj The cat napped **DET NOUN VERB**	[napped]	[.]	$\text{shift}_{\text{PUNCT}}$
7	det nsubj The cat napped . **DET NOUN VERB PUNCT**	[napped .]	[]	$\text{right}_{\text{punct}}$
8	det nsubj punct The cat napped . **DET NOUN VERB PUNCT**	[napped]	[]	root
9	root det nsubj punct The cat napped . **DET NOUN VERB PUNCT**	[]	[]	-

Table 3.1: A shift-reduce process to create a tagged dependency tree of the sentence *The cat napped.*

3.1.2 Transitions, Oracles, & the Shift-Reduce Algorithm

The shift-reduce algorithm builds a tree-structure through the use of two working components: a *stack* and a *buffer*. In the simplest case, there are two actions, or *transitions*, that can be taken to modify these components.

Initially, all words in an utterance to be parsed reside in the buffer, but a *shift* action moves the first element of the buffer to the top of the stack (see the changes between rows 1-2, 2-3, 4-5, and 6-7 of Table 3.1). When there are at least two elements on the stack, tree-building can begin with a *reduce* action that simultaneously creates a relation between the top two elements of the stack and pops the dependent element in the relation off of the stack (see the changes between rows 3-4, 5-6, and 7-8 of Table 3.1). A sequence of such actions can build a full syntactic tree for an utterance.

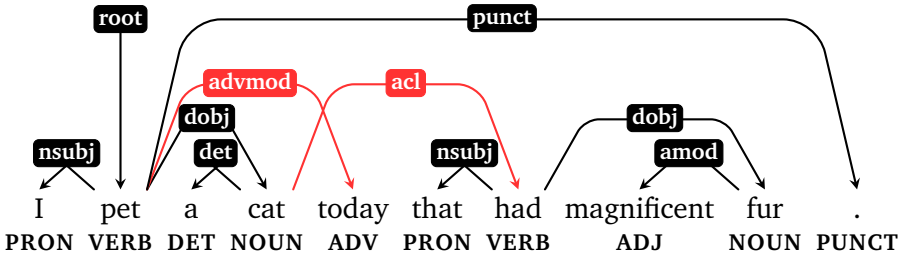
The system of transitions used by a particular parser can vary in a number of ways. In Table 3.1, the *reduce* steps are actually specified as *right* or *left* based on the direction from head to dependent for the arc being created. This much is universal, but what *can* vary is the coupling of arc creation and dependent reduction (i.e., popping the dependent from the stack).

Among the earliest transition-based dependency parsers, Nivre (2003) and Nivre and Scholz (2004) use *arc-eager* systems, which have separate transitions for *left-arc*, *right-arc*, and *reduce*. This leads to longer transition sequences and the possibility of ill-formed trees, but can also lead to better performance. More like the example from the table, Yamada and Matsumoto (2003) and Nivre (2004) are early parsers that instead use an *arc-standard* system, where the *left* and *right* transitions inherently include the reduction step, as well. An *arc-hybrid* system (Kuhlmann et al., 2011) looks much like an arc-standard system, but *left* arcs are made when the head is in the buffer.

Transition systems can also be coupled or decoupled with arc relation and POS tag labels. Most parsers couple the *left* and *right* transitions with arc relation labels, as does the example in Table 3.1, which means there are actually as many possible *left* and *right* transitions as there are relation labels. Some multitask systems of parsing and tagging, again like the parser from the table, also couple POS tags with *shift* transitions (Hatori et al., 2011; Bohnet and Nivre, 2012). Among multitask parsers that *de*couple POS tagging from *shift* transitions, some add in the labeling step as its own separate transition (Yang et al., 2017) while others predict labels independent of transitions entirely (Zhang and Weiss, 2016).

The last major differentiation among transition systems is in the handling of *non-projective dependencies*. Some natural language phenomena require discontinuous syntactic structures that break the rules of well-formed trees, as in Example 4:

(4)



The crossing of the *advmod* and *acl* arcs, which are colored red for emphasis, is the visual indicator of non-projectivity. To make such constructions possible for transition-based parsers, some include a *swap* transition that removes the second-to-top element from the stack and puts it at the front of the buffer (Nivre, 2009; Nivre et al., 2009). This reordering makes it possible to build arbitrary, non-projective dependency graphs, though the complication this creates is sometimes avoided by reordering words in a preprocessing step (Zhang and Weiss, 2016; Yang et al., 2017).

While there may be more than one transition sequence that creates the desired parse tree for a given utterance, the sequence that's provided to a model during training is called the *oracle*. In general, a single, optimal *static* oracle is generated from the dependency annotations for a given utterance. Some work has also explored *dynamic* oracles that use some suboptimal transition sequences in training, though this is more viable for arc-eager systems than arc-standard systems (Goldberg and Nivre, 2012, 2013).

Transition systems and oracles are particularly important to the current work because I use them not only for parsing but also for deriving corpus metrics. It's possible but not straightforward to compare dependency structures directly. However, since I already derive oracles for each utterance in a corpus in order to train parsers, these oracles can be exploited as sources of syntactic information. Much like the POS n-grams of Çöltekin and Rama (2018) and von Prince and Demberg (2018), which capture aspects of typological word order, n-grams over oracle transition sequences ought to capture word order and more.

To get the most out of these dual-use oracles, I combine the simplicity of an arc-standard system with the flexibility of non-projective *swap*ping and the extra information of multitask POS tagging. I end up using slightly different oracles for metrics than for training simply due to the particulars of each use case. The oracles used for metric creation couple arc labels with *reduce* transitions and POS tags with *shift* transitions, while both types of labels are predicted separately from transitions in parsing. Either way, the overall set of transitions are *shift*, *left*, *right*, *swap*, and *root*. I also use purely optimal oracles for metrics while adapting dynamic oracle creation for parser training; I explain all of this in more detail in Chapters 4 and 5.

 # sent_id = GUM_fiction_garden-5
 # text = More rain would come.

ID	FORM	LEMMA	UPOS	XPOS	FEATS	HEAD	DEPREL	DEPS	MISC
1	More	more	ADJ	JJR	Degree=Cmp	2	amod	2:amod	_
2	rain	rain	NOUN	NN	Number=Sing	4	nsubj	4:nsubj	_
3	would	would	AUX	MD	VerbForm=Fin	4	aux	4:aux	_
4	come	come	VERB	VB	VerbForm=Inf	0	root	0:root	SpaceAfter=No
5	.	.	PUNCT	.	_	4	punct	4:punct	_

Table 3.2: An example of an annotated sentence from the English GUM corpus (Zeldes, 2017).

3.2 Universal Dependencies

3.2.1 The UD Framework

Universal Dependencies (UD) is a popular framework for morphosyntactic annotation across natural languages (Nivre et al., 2016; de Marneffe et al., 2021). It puts forth schemes for both universal and language-specific labeling, including dependency parsing as well as POS tags and even some morphological labels. A continuously-updating list of corpora are built upon the framework and are freely available through the UD website[1]. Most recently, as of the version used in the current work, there were 202 corpora across 114 languages.

All UD corpora are formatted in a version of the CoNLL-X format (Buchholz and Marsi, 2006) called CoNLL-U. Table 3.2 shows the annotations for a sentence from an English corpus in UD. All corpora are divided into sentences, which are delimited by newlines but also marked by the ID column starting from 1. Sentences may be fragmentary utterances, as opposed to the syntactic sense of a sentence, but are generally one syntactic sentence at maximum.

Each word in the sentence is marked with its index within the sentence (ID), its surface form (FORM), the lemmatization of its surface form (LEMMA), its universal and language-specific POS tags (UPOS & XPOS), its morphological features (FEATS), the index of its dependency head word (HEAD), the relation label of its dependency arc (DEPREL), the enhanced dependency graph (DEPS), and miscellaneous annotations that minimally include whitespace information (MISC).

[1] https://universaldependencies.org

The ready availability and cross-linguistic coverage of UD make it a common platform for benchmarks[2] and shared tasks, like the aforementioned CoNLL shared tasks for two consecutive years (Zeman et al., 2017, 2018) and the Shared Task on Measuring Language Complexity (Berdicevskis and Bentz, 2018). Having processed any one corpus in UD, it's just as easy to process any other UD corpus. Familiarity with a particular corpus or its language isn't prerequisite to using it.

In the current work, I use only some of the annotations for parsing and metric calculation. Unlike the entries in Zeman et al. (2017, 2018), which were required to take raw, untokenized text as input and perform word tokenization in addition to parsing and labeling tasks, I take for granted both sentence *and* word tokenization, purely for the sake of simplicity. I use the character-level information of the surface forms of each word, and this plus whitespace information are the only parser inputs; UPOS labels and arcs (both heads and relation labels, omitting language-speficic labels) are learned by the models. The program that calculates corpus metrics has access to all of this information – tokenization, surface forms, UPOS, heads, and relations. All other annotations, including lemmas and morphological features, are ignored. This is largely just a design choice; I wanted to build parsers that were reliant only on tokenized, character-level input, with additional information learned by the models rather than provided. POS tagging is the extent of multitask labeling done by the parsers, again mostly just for the sake of simplicity.

3.2.2 Corpora in UD

The broad coverage of languages across UD corpora is one aspect that makes UD a strong platform for exploring multilingual-focused multi-source modeling, but I'm interested in some specific language corpora, as well. One type of cross-corpus relationship I want to explore is that of two corpora of closely-related languages that use entirely different writing systems. Several language pairs meet these criteria and have some representation within UD, including Turkish (Sulubacak et al., 2016; Türk et al., 2020) and Uyghur (Aili et al., 2016), as discussed in Chapter 1; Kurmanji (Gökrmak and Tyers, 2017) and Persian (Seraji et al., 2016); and Upper Sorbian and Russian.

[2] http://nlpprogress.com/english/dependency_parsing.html

Another dynamic of interest is that of an orthographic-syntactic similarity tradeoff. This could be as extreme as the trio of *Uyghur-Turkish-Persian*, – Uyghur, being a Turkic language written in an Arabic script, having high syntactic similarity and zero orthographic similarity to Turkish vs. minimal syntactic similarity and moderate orthographic similarity to Persian – but is generally more subtle. *English-Dutch-French* is a more subtle case of interest, as is Turkish-German code-switching (Çetinoğlu and Çöltekin, 2019) with pure German and pure Turkish – the first languages being the focal points, the second languages being more syntactically similar to the focal languages, and the third languages being more orthographically similar. Turkish-German code-switching is interesting in a broader sense as a contact language, as is Maltese (Čéplö, 2018) to a certain extent – the latter being an Italian-influenced Semitic language.

Very low-resource languages like Kurmanji and moderately low-resource languages like Uyghur are also of interest for their levels of coverage (these designations being specific to their coverage within UD). Low-resource languages simply stand to benefit the most from successful multi-source training experiments. Very high-resource languages like English are also of interest here, however, and not just as candidate supplemental corpora; with larger corpora like the English GUM corpus (Zeldes, 2017), it's possible to look at subsamples of the training sets to see whether high-resource languages behave like truly low-resource languages at smaller sample sizes.

In general, I focus on Turkish and English as representative of typological extremes in terms of morphology and syntax. English being one of the highest-resource languages and Turkish being the highest-resource Turkic language, it's particularly interesting to look at different multi-source cases where these are among the candidate supplemental corpora.

The full set of corpora used throughout the experiments, all of them taken from the UD 2.8 collection (Zeman et al., 2021), are listed in Table 3.3. In addition to the above cases of interest, other language corpora are included both to fill out the typological space and to cover enough of the same test sets as Zeman et al. (2018) that the shared task's scoreboards can serve as a baseline. My aim is that there are enough data points for metric analysis to be significant, and that these data points are typologically diverse enough to make the results linguistically satisfying, as well.

Corpus	Language	Genus	Family	Writing System	Train size
ar_padt	Arabic	Semitic	Afro-Asiatic	Arabic	223K
bxr_bdt	Buryat	Mongolic	Mongolic	Cyrillic	153
cs_cltt	Czech	Slavic	IE	Latin	27K
cs_fictree	Czech	Slavic	IE	Latin	133K
de_gsd	German	Germanic	IE	Latin	263K
en_gum	English	Germanic	IE	Latin	101K
en_lines	English	Germanic	IE	Latin	57K
en_partut	English	Germanic	IE	Latin	43K
fa_seraji	Persian	Iranian	IE	Arabic	121K
fi_ftb	Finnish	Finnic	Uralic	Latin	127K
fi_tdt	Finnish	Finnic	Uralic	Latin	162K
fr_partut	French	Romance	IE	Latin	24K
fr_sequoia	French	Romance	IE	Latin	50K
hsb_ufal	Upper Sorbian	Slavic	IE	Latin	460
it_isdt	Italian	Romance	IE	Latin	275K
it_partut	Italian	Romance	IE	Latin	48K
ja_gsd	Japanese	Japanese	Japanese	Kanji, Kana	168K
kk_ktb	Kazakh	Northwestern	Turkic	Cyrillic	529
kmr_mg	Kurmanji	Iranian	IE	Latin	242
ko_gsd	Korean	Korean	Korean	Hangul	56K
ko_kaist	Korean	Korean	Korean	Hangul	296K
lt_alksnis	Lithuanian	Baltic	IE	Latin	47K
lt_hse	Lithuanian	Baltic	IE	Latin	3K
lv_lvtb	Latvian	Baltic	IE	Latin	192K
mr_ufal	Marathi	Indic	IE	Brahmic	2K
mt_mudt	Maltese	Semitic	Afro-Asiatic	Latin	22K
nl_lassysmall	Dutch	Germanic	IE	Latin	74K
olo_kkpp	Livvi	Finnic	Uralic	Latin	144
pl_lfg	Polish	Slavic	IE	Latin	104K
pl_pdb	Polish	Slavic	IE	Latin	281K
qtd_sagt	Turkish German	Code switching	Code switching	Latin	10K
ru_gsd	Russian	Slavic	IE	Cyrillic	74K
sv_lines	Swedish	Germanic	IE	Latin	55K
sv_talbanken	Swedish	Germanic	IE	Latin	66K
ta_ttb	Tamil	Southern	Dravidian	Brahmic	6K
te_mtg	Telugu	South Central	Dravidian	Brahmic	5K
tr_boun	Turkish	Southwestern	Turkic	Latin	98K
tr_imst	Turkish	Southwestern	Turkic	Latin	37K
ug_udt	Uyghur	Southeastern	Turkic	Arabic	19K
zh_gsd	Chinese	Sino-Tibetan	Sino-Tibetan	Hanzi	98K

Table 3.3: Basic stats on the full set of corpora used for calculating metrics and training parsers in the current work. All corpora come from the UD 2.8 collection (Zeman et al., 2021). Language family information comes from listings on the UD website, which in turn mostly references WALS online (Haspelmath et al., 2005; Dryer and Haspelmath, 2013). Source genres for each corpus are listed in Appendix A (Table A.6).

CHAPTER 4
Models

4.1 Modeling Ethos

All models trained for the current work are neural, lexicalized, transition-based dependency parsers. In general, they should have access to representations of the data that allow them to pick out the same patterns in the data as the predictive metrics.

In particular, the models start from character-level representations of each word, sans pretraining, and jointly predict POS labels rather than using them as input. The training corpora themselves are the sole factor in performance, and the characters (i.e., the orthography) get to play a central role. The joint, or multitask, learning of POS tagging gives the models access to these labels at training time without allowing them to necessarily overshadow the text itself.

The models should perform competitively with those from the literature that have been trained and tested on the same corpora, but it's most important that they perform well enough for comparisons between them to be meaningful. All models have the same architecture, varying only inasmuch as the character inventories of the training corpora vary. Training is as comparable as possible between and among single-source and multi-source conditions.

With the architecture and training of the models as controlled as possible, I can not only isolate the corpora used for training as the primary experimental variable, but also trust that the corpus-derived metrics can capture consistent patterns between the corpora, task, and models. Between character-level input and predictions of POS labels and dependencies, the models can be reasonably expected to capture similar orthographic and syntactic patterns to the metrics in Chapter 5.

4.2 Data Processing

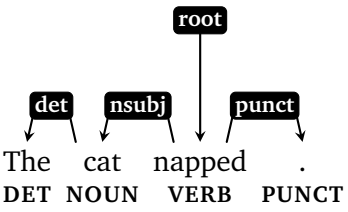
Models use only characters as input – but also, unlike some models from the literature, gold-standard word and sentence tokenization (or segmentation). This is solely for the sake of simplicity.

Since the models are trained and tested solely on UD corpora, which are annotated like the example in Table 3.2 from Chapter 3, the main data processing task is to generate the oracle transition sequences from each sentence's dependency annotations. The oracles used to train the parsers, like those used for calculating metrics, are arc-standard (Yamada and Matsumoto, 2003; Nivre, 2004). With the inclusion of a *swap* transition to account for non-projective dependencies (Nivre, 2009; Nivre et al., 2009), the set of transitions that make up every oracle are *shift*, *left*, *right*, *swap*, and *root*. Because arc relation labels and POS tags are predicted in a manner decoupled from the transitions, there aren't any further subdivisions beyond those five – for the parsers, that is. The oracles used in metric calculation *do* couple labels with transitions (see Chapter 5).

In an unpublished work that originated the family of models used in the current work, the decoupling of labels from transitions yielded slightly improved parsing performance. An additional strategy for optimal oracle usage in the current work is inspired by the dynamic oracles of Goldberg and Nivre (2012, 2013). Goldberg and Nivre (2013) assert that dynamic oracles, which involve exploratory training with suboptimal transition sequences, are not viable in arc-standard systems. This comes down the impossibility of proving whether a correct parse is possible after a suboptimal decision. To get around this, I exploit the *swap* transition, which allows for probabilistic selection of suboptimal sequences that can, if nothing else, be made viable through reordering.

Take, for example, a very simple dependency parse:

(5)



The optimal (label-coupled) transition sequence for the sentence in Example 5 would be $shift_{DET}$ $shift_{NOUN}$ $left_{det}$ $shift_{VERB}$ $left_{nsubj}$ $shift_{PUNCT}$ $right_{punct}$ *root*. Instead, the parser might skip the $left_{det}$ step at first, leaving *the*, *cat*, and *napped* all on the stack with no way to make the *det* arc. After two *swaps* in a row, however, *the* and *cat* are back to the top two positions of the stack, meaning the arc can be made and the correct parse still reached – in a roundabout way. The final sequence would be $shift_{DET}$ $shift_{NOUN}$ $shift_{VERB}$ $shift_{PUNCT}$ $right_{punct}$ *swap swap* $shift_{DET}$ $shift_{NOUN}$ $left_{det}$ $right_{nsubj}$ *root*.

This can get messy in a few ways, including the possible double-assigning of POS labels (which is a problem with any use of the *swap* transition). Since the transitions are decoupled, POS labels are only predicted the first time a word is shifted onto the stack. To make sure transition sequences are still mostly optimal and not overly long loops of *swap*ping, I probabilistically select between *shift*ing and arc creation at points where the arc creation step would be optimal; the probabilities are tuned so that there's an average of one suboptimal *shift* per sentence, and once it becomes necessary to *swap*, the rest of the transitions are all optimal. Preliminary experiments suggest that generating oracles like this at each training step improves performance over static oracle training.

Beyond oracle creation, data processing is fairly straightforward. I use the raw surface forms, UPOS labels, and arc relation labels as-is – though the relation labels are stripped of language-specific additions when a corpus includes them. The use of compositional character models for word representations renders any text normalization (e.g., removal of capitalization) unnecessary.

4.3 Neural Architecture & Implementation

4.3.1 Components & Parameters

All parsers in the current work are built in PyTorch (Paszke et al., 2019) and follow the tradition of Chen and Manning (2014) and the neural transition-based dependency parsers that followed it, especially multitask (or *joint*) POS-tagging variants (Zhang and Weiss, 2016; Yang et al., 2017). Philosophically, the parsers stem most directly from those trained by Ballesteros et al. (2015), who opt for minimalistic models that exclude auxiliary data (i.e., pretrained word representations), beam search, preprocessing of non-projective dependencies, and explicit feature engineering.

I specifically employ a *stack-LSTM* model similar to those of Dyer et al. (2015) and Ballesteros et al. (2015). In addition to using *long short-term memory* (LSTM; Hochreiter and Schmidhuber, 1997) cells to model the stack and buffer, the models use character-level *convolutional neural networks* (CNNs; LeCun et al., 1990) to compose word-level embeddings from the character level – similar to the models in Santos and Zadrozny (2014) and Yu and Vu (2017).

Ballesteros et al. (2015) actually use bidirectional character LSTMs (BiLSTMs; Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005), but word representations learned via CNN for character composition have proven successful in both part-of-speech tagging and dependency parsing tasks. Santos and Zadrozny (2014) use character-CNNs to learn word representations for POS tagging in both English and Portuguese, and Yu and Vu (2017) use them in dependency parsing of several morphologically complex languages. The latter compare CNN-based composition to a BiLSTM-based composition baseline (as well as non-compositional, word-level embeddings). In almost every case, CNN-based composition outperforms its BiLSTM (and word-level) counterpart.

Zhai et al. (2018) do a similar comparison of word representations in named-entity recognition. They find that CNN-based composition performs at least as well as LSTM-based composition and trains only 25% slower than word-level embedding, while LSTM-based composition takes twice as long to train as word-level embedding.

In all, prior literature and preliminary experiments point to character-CNN-based word representations being optimal for both training time and test performance. I minimize test time by caching the composed representations of common words, as well.

Some basic parameters of the architecture shared by all of the models are shown in Table 4.1. Throughout the networks, hidden layers (i.e., those after non-output linear functions) are run through a *Rectified Linear Unit* (ReLU) nonlinearity (Hahnloser et al., 2000). All *Recurrent Neural Network* (RNN) cells are LSTM cells and have only a single layer of depth. Embedding sizes are fixed so that the only source of model size variation is in the number of character embeddings.

The character CNN component is modeled after that of Yu and Vu (2017), with four convolutional kernels of varying size, to which a max-over-time pool followed by a linear layer are then applied to yield a fixed-size representation of each word. This base representation is used in multiple ways to represent the buffer, stack, and composition as parsing progresses.

4.3.2 Process & Diagrams

Parsing happens step-by-step – or transition by transition. At each stage in parsing a sentence, the configurations of the stack and buffer are used to create a representation of the *parser state*.

PARAMETER	VALUE
Nonlinearity	ReLU
RNN cell type	LSTM
RNN depth	1
RNN state size	256
POS embedding size	32
Arc embedding size	32
Character embedding size	64
Base word hidden size	128
Composed word hidden size	256
Convolution kernel sizes	3, 5, 7, 9

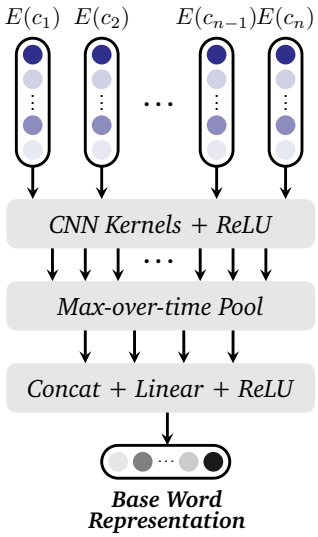
Table 4.1: Basic parameters common to the architecture of all trained models.

Figure 4.1 shows how the initial word representations are created, as well as the composition[1] steps that occur at each *shift* and reduce step, similar to Dyer et al. (2015) and Ballesteros et al. (2015). Whitespace characters (or implicit end-of-word markers if there is no whitespace) are included as the first and last character embeddings for a word, and the embeddings for POS and arc relation labels (composed upon *shift*ing and reduction steps, respectively) are those of the gold labels at training time and predicted labels at test time.

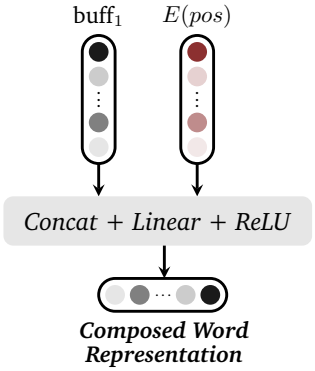
Figure 4.2 shows the stack and buffer LSTMs as a part of the network for generating the output probabilities of transitions, arc relation labels, and POS tags. Each word in the stack and buffer windows starts with its vector representation as yielded by the processes of Figure 4.1. These representations are then fed into the stack and buffer LSTMs, respectively. (When possible, the LSTM states and outputs are cached for efficiency rather than fully recalculated at every step.)

The top two outputs of the stack LSTM and the first two elements of the buffer BiLSTM are concatenated and run through a linear layer to yield the *parser state* representation. This vector is first used for the linear prediction of the next transition. The same vector is then used to predict the POS tag of the word being *shift*ed or the relation label of the arc being created, depending on the predicted (or, during training, gold) transition.

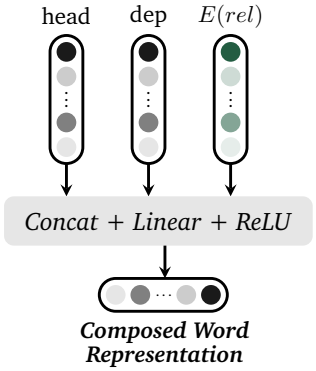
[1]The use of *composition* here is a little muddled, since there is both the character composition used to create the word representations as well as the subsequent composition of those word representations with tag labels, arc relation labels, and other word representations. For the purposes of the architecture, *composed* representations refer to the latter sense.



(a) The component network for constructing the initial character-based word representations.



(b) The component network for composing the base word representation with its POS tag when the word is shifted onto the stack.



(c) The component network for composing the new representation for a head word at each reduction step. There are actually two such linear functions, based on whether the reduction is left- or right-headed.

Figure 4.1: The three parts of the model used for the composition of word-level representations. Each $E(x)$ represents an embedding.

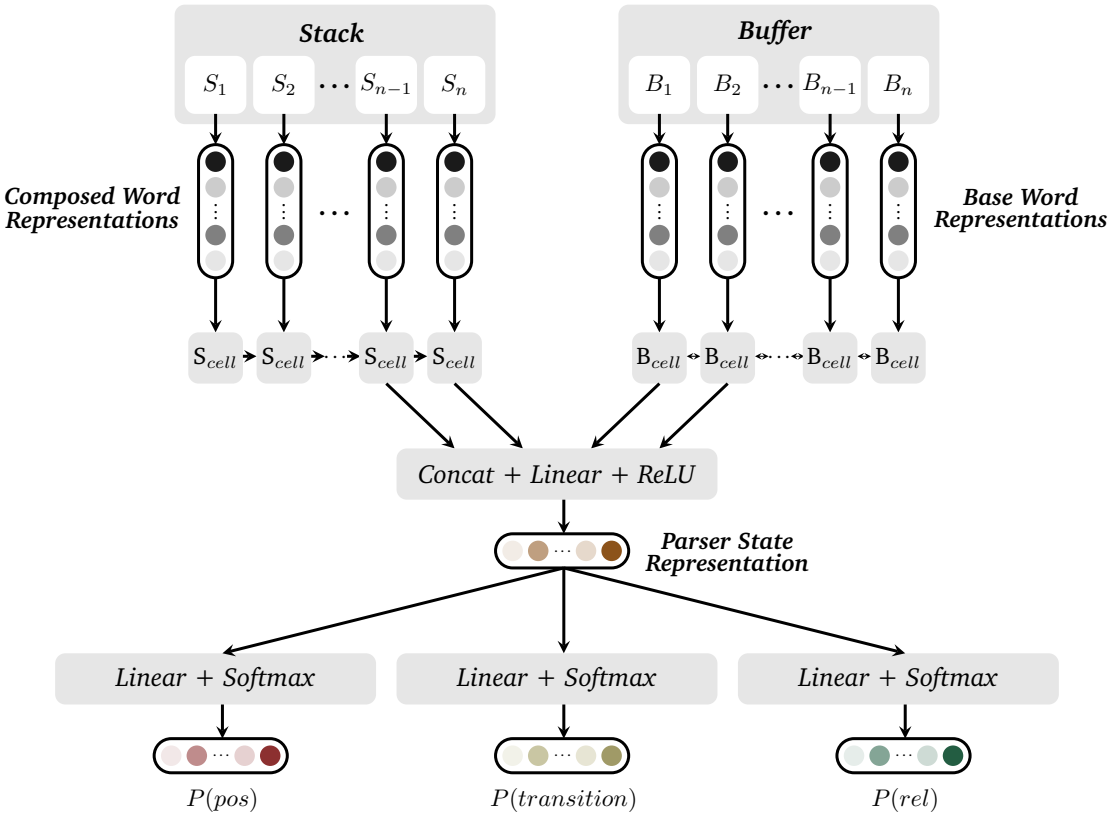


Figure 4.2: The part of each model that creates the main parser state representation and subsequent output probabilities. Each $P(x)$ represents a probability.

Finally, the predictions of the network in Figure 4.2 are used to get the embeddings for POS tag and arc relation labels that are then used to create new, composed representations of *shift*ed or head words via the networks of Figure 4.1. This feedback loop continues until the stack and buffer are empty and all arc relation and POS tag labels predicted.

4.4 Training Strategies

4.4.1 Single-Source vs. Multi-Source Training

The most significant design decision regarding training the parsers is in balancing the difference between single-source and multi-source conditions. Single-source conditions are simple, – train on the single target corpus all the way through – but questions arise for multi-source conditions.

The default strategy would be naïvely concatenative training, where target and supplemental training corpora are simply treated as a single corpus for the duration of training. Preliminary experiments confirm, however, that training with a fine-tuning stage – where only data from the target training set are included for the latter portion of training steps – yields consistent improvement over the naïve approach. My goal is to train the best multi-source models possible within the constraints of the research questions, but the fine-tuning approach runs the risk of making it so that the multi-source models are no longer directly comparable to the single-source models – namely, if the number of training steps (relative to the size of the training corpora) are too different.

While I set out to include both naïve and fine-tuned multi-source models in my analysis, I ultimately use only fine-tuned training of multi-source models. Chapter 7 includes some of the initial experiments comparing the two strategies, but because fine-tuned performance is so consistently better and the difference from single-source training minimal, I determine that the fine-tuned models are sufficient for analysis. This means that the models are not intended to be multilingual in practice. The naïve approach yields reasonable models for the target *and* supplemental corpora, whereas the fine-tuning approach yields models that, while still *viable* as parsers for the supplemental corpora, are unilaterally worse in that respect than single-source models. As the research questions are about performance on target corpora alone, this is an acceptable tradeoff.

4.4.2 Hyperparameters

Table 4.2 shows the full array of hyperparameters used across the models, both single- and multi-source. I use the term "epoch" loosely, as training is done by randomly sampling the training corpora to get each batch rather than iterating over each element of the training set exactly once. An "epoch" ends up being three times the size of the training set, divided by the batch size; changes in batch size are what primarily distinguish one "epoch" from the next.

For multi-source conditions, the fine-tuning stage simply means that, during the final three "epochs," only data from the target training corpus are sampled for the training batches. Prior to that, batches are sampled from the concatenation of the target and supplemental training corpora. This is the only difference between the treatment of single-source and multi-source models.

PARAMETER	VALUE
Loss function	Softmax cross-entropy
*Epochs**	10
Batch size	$(3/2)^{epoch}$
Optimizer	AdamW
Learning rate	5e-4
Weight decay	1e-2
Initial dropout rate	0.10
Final dropout rate	0.19

Table 4.2: The training hyperparameters used across all trained models.

I use the default PyTorch settings for the AdamW optimizer (Loshchilov and Hutter, 2019), which is a variant of the Adam optimizer (Kingma and Ba, 2015). It decouples the weight decay, thus working best with a much higher weight decay than standard Adam, and achieves empirically better generalization.

Rather than using learning rate annealing (lowering learning rate over the course of training to facilitate convergence; e.g., used in Dyer et al., 2015; Ballesteros et al., 2015), I increase the batch size over the course of training, which yielded better results than the more traditional annealing in preliminary experiments. Starting with a batch size of 1 and multiplying the batch size by a constant with each subsequent epoch (at which point the optimizer is also reset) proves both simple and effective for quick convergence. According to Smith et al. (2018), increasing the batch size is mathematically equivalent to decreasing the learning rate, though I find the former more effective in practice.

The loss function is cross-entropy with a softmax activation, similar to the log-likelihood loss used in some literature (e.g., Dyer et al., 2015). Each training step comprises a sort of pseudo-batch, optimizing the summed losses across each transition and additional (POS and arc relation) label in however many utterances are in the batch.

It's not "true" batching, which normally means computations are done on elements of a batch in parallel as a single matrix/tensor, but the strictly sequential nature of the parser makes this impossible. Thus, a "batch" here simply refers to all of the computations over which the gradient is calculated for a single optimizer step.

To prevent overfitting, I employ dropout (Srivastava et al., 2014) with several strategies in addition to the original method. Dropout is the zeroing-out of probabilistically-selected elements of a given vector in the network to facilitate generalization instead of memorization. A *mask* vector is generated to comprise only 1's and 0's based on the *dropout rate* hyperparameter. A unique multiplicative mask is used for each dropout layer in a network, and in traditional dropout the masks are re-generated with each use of a vector.

Based on preliminary experiments, the best convergence for the current models comes from using *curriculum dropout* (Morerio et al., 2017), wherein dropout rate increases over the course of training. The most effective and least computationally costly strategy is, in particular, *variational curriculum dropout* with a linear schedule (O'Neill and Bollegala, 2018), wherein dropout increases at a constant rate (after each "epoch") and each dropout layer's mask is only computed once per training step.

Variational dropout (Gal and Ghahramani, 2016) is effective for applying dropout to RNNs, as it uses the same mask for a given connection across all time steps in a sample. It also makes sense for the non-recurrent weights of the current parsing architecture; all dropout layers are used multiple times per optimizer step in a strictly sequential way. This way of computing dropout masks is at least as effective as simple dropout and significantly more efficient for larger batch sizes.

All hyperparameters were selected through preliminary experiments to create similarly strong convergence and minimal overfitting for corpora big and small – both in terms of training corpus size (i.e., high- vs. low-resource) and input character inventories (e.g., Japanese vs. English).

Part II

Contribution

CHAPTER 5
Metrics

5.1 Basis & Goals

The primary utility of the corpus metrics put forth here is in predicting the improvement of multi-source models over single-source ones. They should accomplish this using only the relevant corpora (including annotations), without the need for external data, nor the need to train every model first.

There are additional desiderata for the metrics, however, given both the dependency parsing testbed and my hypotheses about what it takes to make predictions about language data. I hypothesize that multiple dimensions of similarity between target and supplemental corpora contribute to the efficacy of a resultant multi-source model. These dimensions include patterns, in linguistic terms, both orthographic and syntactic; in data terms, both for inputs and for labels.

I also hypothesize that the intrinsic sparsity of the target training corpus determines how beneficial a similar-enough supplemental corpus will be. As target resources grow larger, supplemental corpora must be larger and more similar in order to make a contribution – until, after a certain point, even additional data of the same language will do little to improve over the single-source baseline. Most of this follows intuitively from an understanding of machine learning and language data, but the challenge at hand is in capturing and proving these patterns.

Another consideration is in regard to generalizability of the metrics and their application. I explore only those metrics that capture patterns I can reasonably expect the parser models described in Chapter 4 to capture, as well. Otherwise, while they may be interesting metrics, they'd be unlikely to confer much benefit in predicting model performance. This connection between models and metrics is arguably more important than the particular metrics for future work in this vein.

Of course, for these experiments, it *is* necessary to train the relevant models as data points for testing the metrics. Most "predictions" use the same data points used to tune the metrics, meaning the evaluations may be optimistic, but the simple predictive methods are at low risk of overfitting.

In this chapter, I describe the foundations both for the corpus metrics themselves and for the predictive methodology built upon them. I begin by establishing what the metrics should even be predicting – and how – before getting into the particular facets of the data being measured and the calculations being done. The metrics include both intrinsic, *intra-corpus* metrics, with the goal of capturing data sparsity; and pairwise, *cross-corpus* metrics, with the goal of capturing similarity.

5.2 Evaluation in a Predictive Methodology

5.2.1 Single-Source Evaluation & Prediction

Single-source models act primarily as baselines for multi-source models, but their parsing performance also matters independently. The standard measure of dependency parsing performance is *attachment score* (Lin, 1995; Eisner, 1996b). The *unlabeled* variant (UAS) is the percentage of word tokens for which the parser predicts the correct dependency head; the *labeled* variant (LAS) is similar but also requires the correct dependency (i.e., arc relation) label.

Since all parsers here are multitask POS taggers, that gives three relevant metrics: UAS, LAS, and POS tagging accuracy. In the interest of avoiding numerical overload, I use only LAS for evaluation and analysis, as parsing is the focal task. Where possible, I compare the LAS of the single-source models to the corresponding scoreboards from the shared task of Zeman et al. (2018). This is to get a grounded idea of how effective the parsers are intrinsically – namely, due to the common model architecture and training strategies – and confirm that comparisons between them are indeed comparing models that are as strong as they can be given the training conditions.

I also devise a predictive methodology specific to single-source models, to be carried out in Chapter 6. This connects indirectly to the final predictions of multi-source improvement by helping to measure instinsic corpus sparsity – which I hypothesize is a major factor in the possibility of multi-source improvement for a target corpus. The problem is that the concept of *sparsity* for a corpus is a bit nebulous. Is it solely about how large the corpus is in terms of characters, words, or utterances? Is it directly tied to the performance of a single-source parser trained on said corpus?

I start with the assumption that the size of a target training corpus (namely in terms of the length in word tokens) is directly proportional to the performance of a single-source model trained on that corpus, and that *sparsity* roughly describes this relationship: the smaller the training corpus, the lower the performance. As Cotterell et al. (2018) and Park et al. (2021) find, however, depending on the family of models, morphologically complex languages *can* be inherently more difficult to model even after controlling for various size effects. I consider this a matter of sparsity, as well; more data means better performance, but some languages require more data (as measured by training corpus length) than others to reach comparable performance.

The core question is this: *Is there an intra-corpus metric better than corpus size in predicting single-source performance?* If not, then corpus size should be the sparsity metric applied to the final multi-source improvement predictions; if so, then the strongest metric should take its place for that purpose. I compare a range of metrics, both intra-corpus and a few that are "cross-corpus" between the training and test splits of the target corpus, in terms of which has the strongest correlation (as determined by linear or logarithmic regression) to single-source LAS.

5.2.2 Multi-Source Evaluation & Prediction

For multi-source models, I look not just at model performance but at model *improvement*. That isn't to say basic model performance as measured by LAS is irrelevant. I still compare the multi-source models to Zeman et al.'s (2018) scoreboards where possible to ground the results, in addition to comparing various multi-source conditions for a single target corpus to one another.

Zeman et al. (2018) use *bootstrap resampling* to determine significant differences between models. Bootstrap significance is an appropriate basis for comparing multi-source to single-source performance on a test set. I compute it as defined in Berg-Kirkpatrick et al. (2012) (see Figure 5.1).

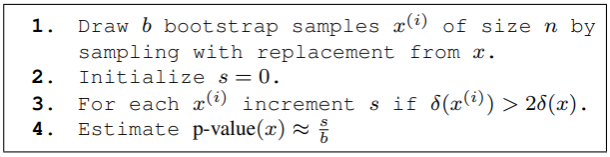


Figure 5.1: The bootstrap procedure as defined in Berg-Kirkpatrick et al. (2012).

Bootstrap tests the hypothesis that, given an observed difference (δ) in model performance on a test set (x), said difference happened by chance. That is, the *null hypothesis* is that the two models in fact have similar performance. A p-value near one means they do seem to have similar performance, based on sampling the test set with replacement; a p-value near zero means the difference is consistent across samples and therefore significant.

As-is, bootstrap significance is not sufficient: in the same way it's possible for models to appear to have a large difference (e.g., 10% LAS) that turns out not to be significant, it's also possible for them to have a small difference (e.g., 0.01% LAS) that *is* technically significant. Thus, I measure *improvement* in LAS scores of a multi-source model (LAS_{MS}) over a single-source model (LAS_{SS}) by combining their absolute difference with bootstrap significance, as in the following formula:

$$Improvement = \begin{cases} (LAS_{MS} - LAS_{SS}), & \text{if } p_{bootstrap} < 0.05; \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

Improvement as derived by Equation 5.1 is equal to the difference in LAS if that difference is statistically significant with 95% confidence; otherwise, it's zero. I define improvement as significant overall if it's greater than 1%. All of this is to derive the measure that the metrics should predict. The question is: *Is there a metric or combination of metrics that can predict whether a multi-source training condition will yield a model that exceeds the significance threshold for improvement?*

Unlike with single-source performance, the expected shape of the distribution makes regression a poor fit for making predictions. Most improvement scores should sit at zero; most multi-source conditions will simply yield no significant difference from single-source conditions. Even extremely dissimilar corpora are unlikely to *degrade* performance significantly. As such, I do not expect the improvement distribution to be linear with respect to any of the metrics. Japanese should have a near-zero similarity to English, while a language like Czech might have around 50% similarity – yet both would likely yield near-zero improvement as supplements to an English corpus.

Thus, instead of comparing regression models, I compare *predictive thresholds*. For each candidate metric, I pick the threshold that best recreates the true significance, with corpora correspoding to models with significant improvement on one side of the threshold and the rest below it.

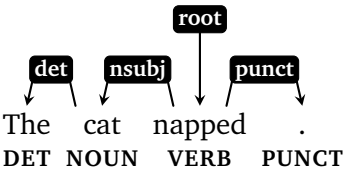
The metrics are then sorted by the predictive power of their best thresholds, and the metric with the best of the best is selected for use in the predictive methodology. Whereas the single-source regression models are compared by R^2, the multi-source metric thresholds are compared by *F1 score* – the harmonic mean of the *precision* and *recall*.

Unlike Ruder and Plank (2017) and Lin et al. (2019), I avoid stronger techniques like *learning-to-rank*. This is in part to focus on the efficacy of each of the metrics independently, and in part because a lot of other techniques don't fit the data very well. Ranking the corpora/models seems more powerful than the situation requires when the majority will be tied for last, so to speak. Ultimately, since the takeaway for future work is in the application of the metrics and not the exact predictions and numbers, the simple thresholding method is preferable in allowing the metrics to remain front-and-center. If they're effective on their own, that lays a foundation for future work.

While the binary significance decision is the primary measure of multi-source improvement, I go further in depth after selecting the best predictive metric for further analysis. This metric-as-methodology is applied to case studies centered on various languages of interest, as discussed at the end of Chapter 3. I also take a less systematic look at predicting the best multi-source conditions for a target corpus (as in Rosa and Žabokrtskỳ, 2015) – noting whether, for target corpora for which multiple multi-source conditions were trained, the best condition has the highest metric value.

5.3 Phenomena, Units, & Features

In order to capture as many facets of each language corpus as possible, I utilize all aspects of the corpora that are available to the models. This means the metrics have a better chance at picking up on the same patterns the models might be learning and thereby serving as stronger predictors, but it also means that, because of the morphosyntactic nature of the UD annotations, the metrics are likely to capture patterns that are linguistically meaningful. Capturing *orthographic* aspects of the corpora involves looking at character-level units; capturing *morphosyntactic* aspects involves looking at the word-level annotations as well as transition sequences. Picking up on orthographic and morphosyntactic *patterns* requires more than just vocabulary lists, especially since all UD corpora use the same labels for annotation; to that end, most of the metrics look at *n-grams*.



$\text{shift}_{\text{DET}}$ $\text{shift}_{\text{NOUN}}$ left_{det} $\text{shift}_{\text{VERB}}$ $\text{left}_{\text{nsubj}}$ $\text{shift}_{\text{PUNCT}}$ $\text{right}_{\text{punct}}$ root

UNITS	N-GRAMS (N=3-6)
characters	#The# #cat# #nap# #app# #ppe# #ped# #napp# #appe# #pped# #nappe# #apped# #napped#
POS	#DET NOUN VERB# #DET NOUN VERB PUNCT#
relations	#det nsubj root# #det nsubj root punct#
POS/rel	#DET/det NOUN/nsubj VERB/root# #DET/det NOUN/nsubj VERB/root PUNCT/punct#
governorships	#$\text{dep}_{\text{nsubj}}$ gov_{root} $\text{dep}_{\text{punct}}$#
transitions	#$\text{shift}_{\text{DET}}$ $\text{shift}_{\text{NOUN}}$ left_{det}# #left_{det} $\text{shift}_{\text{VERB}}$ $\text{left}_{\text{nsubj}}$# #$\text{shift}_{\text{DET}}$ $\text{shift}_{\text{NOUN}}$ left_{det} $\text{shift}_{\text{VERB}}$# #$\text{left}_{\text{nsubj}}$ $\text{shift}_{\text{PUNCT}}$ $\text{right}_{\text{punct}}$ root# #$\text{shift}_{\text{DET}}$ $\text{shift}_{\text{NOUN}}$ left_{det} $\text{shift}_{\text{VERB}}$ $\text{left}_{\text{nsubj}}$# #$\text{shift}_{\text{NOUN}}$ left_{det} $\text{shift}_{\text{VERB}}$ $\text{left}_{\text{nsubj}}$ $\text{shift}_{\text{PUNCT}}$# #$\text{shift}_{\text{DET}}$ $\text{shift}_{\text{NOUN}}$ left_{det} $\text{shift}_{\text{VERB}}$ $\text{left}_{\text{nsubj}}$ $\text{shift}_{\text{PUNCT}}$#

UNITS	UNIGRAMS
words	#the# #cat# #napped# #.#

UNITS	PAIRS
char + arcs	$\text{dep}_{\#\text{the}\#}$+$\text{left}_{\text{det}}$ $\text{gov}_{\#\text{cat}\#}$+$\text{left}_{\text{det}}$ $\text{buf}_{\#\text{the}\#}$+$\text{shift}_{\text{DET}}$
POS + arcs	dep_{DET}+left_{det} gov_{NOUN}+left_{det} dep_{VERB}+root

Table 5.1: A sample English sentence with a (truncated) selection of its derived features, which are then used to build vocabularies and calculate metrics based on those vocabularies.

The entries in the shared task of Berdicevskis and Bentz (2018) serve as the most direct inspiration for the units used by the metrics – Çöltekin and Rama's (2018) and von Prince and Demberg's (2018) POS n-gram distribution perplexities in particular. I build off of the idea of POS n-grams and look at n-gram features based on surface forms of words, word-level annotations, and oracle transition sequences.

Table 5.1 shows some of the n-gram features derived from a simple example sentence. I consider n-grams for all values of n between 3 and 6 – a range selected through preliminary experiments. At an n of 1 or 2, n-grams pick up on highly frequent and minimally meaningful units, like punctuation. I capture these sorts of units by keeping track of unigrams for words only.

The character n-grams, which look only at sequences within words and not across them, serve as the source of orthographic information. In addition to capturing overall spelling patterns, they also capture units that, even if they're not necessarily morphemes, tend to be morpheme-sized. For the example sentence in Table 5.1, the verb *napped* yields character trigram features that include the not-very-informative "app" as well as the much-more-informative "nap" and "ped" – the former capturing the lemma and the latter largely capturing the past-tense suffix.

I do also include whole words as features, but specifically not as n-grams for $n > 1$. They don't add much orthographic information that the character n-grams don't already capture, and it's so rare for corpora to share even trigram word sequences that they don't indicate much beyond pinpointing corpora that are of both the same language *and* the same domain. As unigrams, however, they can potentially capture common words cross-linguistically.

The POS, arc relations, and combination *POS/relation* n-grams are fairly straightforward. The transition n-gram features are similar – they just look at the *oracle* sequences rather than the labels over the words of the sentence. As discussed in Chapters 3 and 4, these oracles differ slightly from those used by the parsers in that the parser's oracles decouple the arc relation and POS tag labels from the reduce and *shift* steps, respectively. The decoupled variant improves model performance, while the coupled variant gives more informative transition n-grams.

The more creative features are those using *governorships* and *pairs* as units. I use the term *governorship* to refer to something akin to phrases or constituents, adapted to dependency graphs. The example from Table 5.1 shows a trigram of the governorship for the root word *napped*, including the word itself and its immediate dependents, each represented by its arc relation label. The concept of governorship n-grams is my attempt to get metrics for the dependency graph directly, as opposed to the indirect method of using transition n-grams.

Finally, the *pairs* combine aspects of the word-level features with the transition-level features. They track the elements of the stack and buffer at each step of the shift-reduce process and pair the character n-gram features and POS labels (not n-grams) with the next transition at that point.

These n-gram features form the basis for a number of intra-corpus and cross-corpus metrics. The n-grams for each category yield vocabulary distributions, enabling entropy and other calculations.

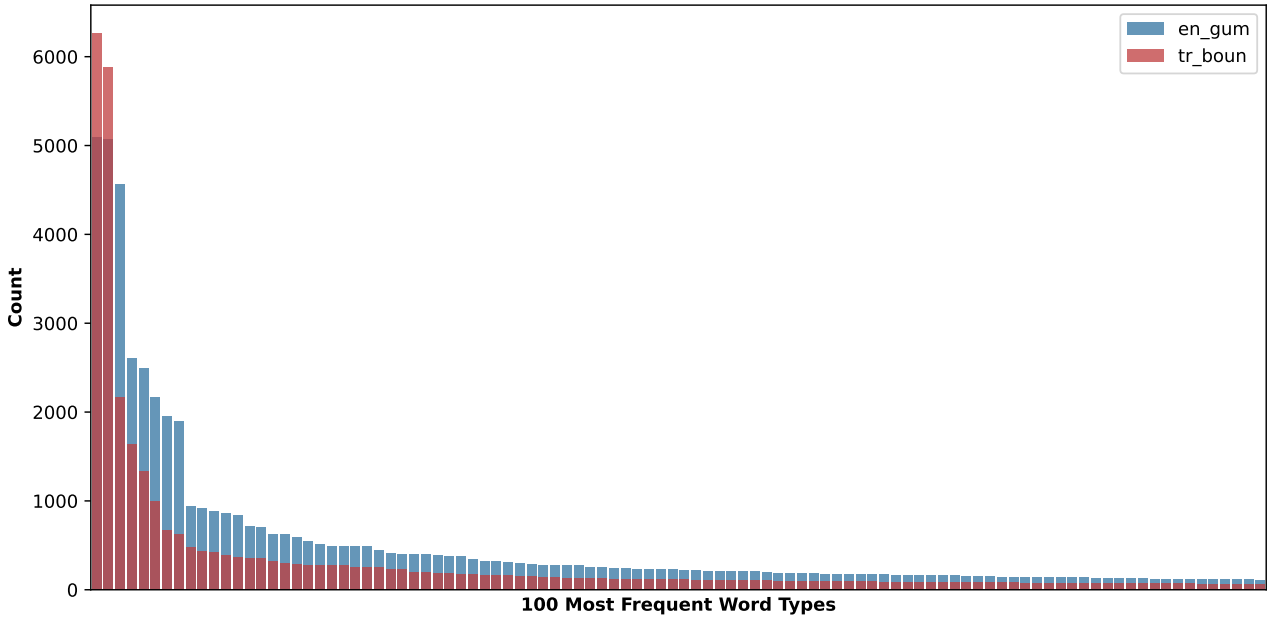


Figure 5.2: The most frequent words in an English (Zeldes, 2017) and Turkish (Türk et al., 2020) UD corpus, both of roughly equal size as measured in number of words.

5.4 Vocabularies, Distributions, & Formulas

While Rosa and Žabokrtskỳ (2015), Çöltekin and Rama (2018), and von Prince and Demberg (2018) look at distributions where the vocabulary is made up of POS n-grams, most discussion of vocabulary distributions in NLP and corpus linguistics center on the word-level vocabulary. The *Zipfian* distribution describes the fact that natural language word frequencies form entirely skewed graphs, wherein there are a few very frequent words and many more infrequent words (Zipf, 1935).

Figure 5.2 shows the word frequency distribution for a Turkish corpus and an English corpus from UD. Both show the telltale Zipfian skew, but it's noteworthy that the shapes are still pretty different despite the corpora having roughly the same number of word tokens. Given that Turkish and English occupy opposite ends of the morphological complexity spectrum, this makes sense; the more complex Turkish has fewer extremely frequent words at the head and a longer tail of rare words than the relatively simple English. Most importantly, this demonstrates that even within the Zipfian definition, it's useful to be able to describe these kinds of differences.

The various n-gram feature sets, when similarly turned into vocabulary frequency distributions, also take on Zipfian shapes. Whether looking at single word or label n-gram distributions, the overall pattern is that, the larger the corpus, the longer the tail – *and* the larger the disparity between the most frequent terms and the average term. Between this and differences like the one observed in the English vs. Turkish distributions based on complexity, vocabulary distributions seem to be good indicators of various kinds of sparsity. They can also be turned into vectors that can be compared across corpora to yield measures of similarity.

Unfortunately, when it comes to characterizing these distributions, traditional measures like *skew* and *kurtosis* effectively only measure corpus size without adding any information; fortunately, several other existing calculations *do* capture the variation within these distributions, if indirectly. Since frequency distributions are based on *types* (x-axis) and *tokens* (y-axis), and since said frequencies are used to calculate probability distributions, metrics like type/token ratio (TTR) and entropy (or perplexity) get at the same information.

Ruder and Plank (2017) and Çöltekin and Rama (2018) calculate metrics including TTR and entropy over words, and Çöltekin and Rama (2018) and von Prince and Demberg (2018) calculate perplexity over POS n-grams, but to my knowledge, this is the first work to apply TTR and a few other metrics to label n-gram distributions, let alone character and transition n-grams.

All calculations start from the following foundations: for a given vocabulary V...

- V is a map of vocab items (*types*) to their observed instances (*tokens*).
- n is the number of types in the vocabulary (equivalent to $|V|$).
- t_i is the ith type in the vocabulary.
- The types in the vocabulary are ordered from most ($i = 1$) to least ($i = n$) frequent.
- $|t_i|$ is the number of observed instances (tokens) for the ith type in the vocabulary.
- $\sum_{i=1}^{n} |t_i|$ is the total number of tokens observed.
- $p(t_i)$ is the probability of observing the ith type in the vocabulary (equivalent to $\frac{|t_i|}{\sum_{i=1}^{n} |t_i|}$).
- μ is the mean type frequency (equivalent to $\frac{\sum_{i=1}^{n} |t_i|}{n}$)
- m is the position of the type with the smallest frequency greater than the mean ($t_m > \mu$, $t_{m+1} \leq \mu$)
- When comparing to another vocab, $\hat{V}$, with the same number of types, the counterpart to t_i is $\hat{t}_i$.

Metric	Formula
Intra-vocab	
Type-Token Ratio	$\frac{n}{\sum_{i=1}^{n} \lvert t_i \rvert}$
Left Mass Percentage	$\frac{\sum_{i=1}^{m} \lvert t_i \rvert}{\sum_{i=1}^{n} \lvert t_i \rvert}$
Entropy	$-\sum_{i=1}^{n} p(t_i) \log_2 p(t_i)$
Type Count	n

Metric	Formula
Cross-vocab	
Cosine Similarity	$\frac{\sum_{i=1}^{n} t_i \hat{t}_i}{\sqrt{\sum_{i=1}^{n} t_i^2} \sqrt{\sum_{i=1}^{n} \hat{t}_i^2}}$
Cross-Entropy	$-\sum_{i=1}^{n} p(t_i) \log_2 p(\hat{t}_i)$

Table 5.2: Various metrics calculated for the different n-gram features shown in Table 5.1.

Based on these definitions, I calculate metrics given each of the n-gram feature sets of the prior section from the formulas in Table 5.2. Of the intra-vocab (or intra-corpus) metrics, *TTR*, *entropy*, and *type counts* are all standard for analyzing corpora at the word level; applying them to the various n-gram distributions is the novel aspect. The only fully original metric is *left mass percentage*, which measures how many tokens have above-average frequencies. The idea is to determine how prominent the head of the distribution is compared to the tail.

The cross-vocab (or cross-corpus) metrics use standard *cosine similarity* and *cross-entropy*. Cosine similarity is preferable to other similarity metrics because it normalizes the vectors, which is very important given that the corpora being compared vary greatly in size. This way, two similar corpora where one is much smaller than the other will still get high similarity scores, as desired.

Cross-entropy is asymmetric, unlike cosine similarity. That is, if the supplemental and target corpora are reversed, cosine similarity will remain the same while cross-entropy will differ. While I don't use the exact same metrics as asymmetric intelligibility literature (Moberg et al., 2007; Frinsel et al., 2015; Kyjánek and Haviger, 2019), cross-entropy makes sense in the current use case, and having some metrics that can capture asymmetrical relations between corpora is desirable.

For both the intra-vocab entropy and the cross-vocab cross-entropy, my probability distributions differ from the standard for entropy-derived metrics in NLP. For example, von Prince and Demberg (2018), in calculating perplexity for POS trigrams, do so in the sequential style of language modeling: probabilities are conditioned on which trigrams follow one another. In contrast, my distributions treat n-grams as frequency-weighted features, without sequential conditioning.

In addition to the 9 n-gram feature families, I also include an *aggregate*, or average, over all of the feature sets for a particular metric. For example, in addition to separate cosine similarity metrics for the character n-grams and transition n-grams of two corpora, there's an average across those two and the other seven to approximate the overall similarity. This is a compromise between only looking at the metrics individually and learning an optimal combination for prediction. Adding them up without a weighting scheme gives a basic sense for what they can accomplish together.

It's worth noting that, unlike Park et al. (2021), I don't control for any length effects – e.g., I use raw TTR rather than a fixed-window version. I do, however, try to make TTR in particular less brittle to the effects of mere repetition. That is, a low TTR is typical for a large natural language corpus, but an artificially low TTR could occur if a small corpus repeated the same few words over and over. My simple solution is to only count unique sentences toward the frequency distributions.

Beyond the n-gram distribution metrics, I calculate a few others, including some basics:

- Corpus length in terms of utterances, words, & characters (tokens)
- Word n-grams per word & character n-grams per word (types)
- Characters per word & words per utterance (tokens)

I also explore some metrics based on dependency graph distances and non-projective dependencies, including a few more specific types of non-projective dependencies (namely *non-planar* and *ill-nested*) as well as the *degree* of a graph (Kuhlmann and Nivre, 2006; Havelka, 2007).

Informally, ill-nested graphs are a type of non-planar graph, which are a type of non-projective graph. Non-planar dependency graphs have crossing dependency arcs (which is not inherently necessary for non-projectivity; non-projectivity just means a node dominates a discontinuous substring of the sentence) and ill-nested dependency graphs contain interleaving disjoint subtrees – ones where two subtrees, where neither root dominates the other, contain crossing arcs.

In theory, more non-projective constructions would make corpora inherently harder to parse. Dependency graph-related metrics include, specifically:

- Swaps per utterance
- Percentage of utterances with non-projective, non-planar, & ill-nested trees
- Average dependency depth (distance from root) & number of dependents per governing node
- Average graph distance from sequential neighbor & sequential distance from dependent to governor

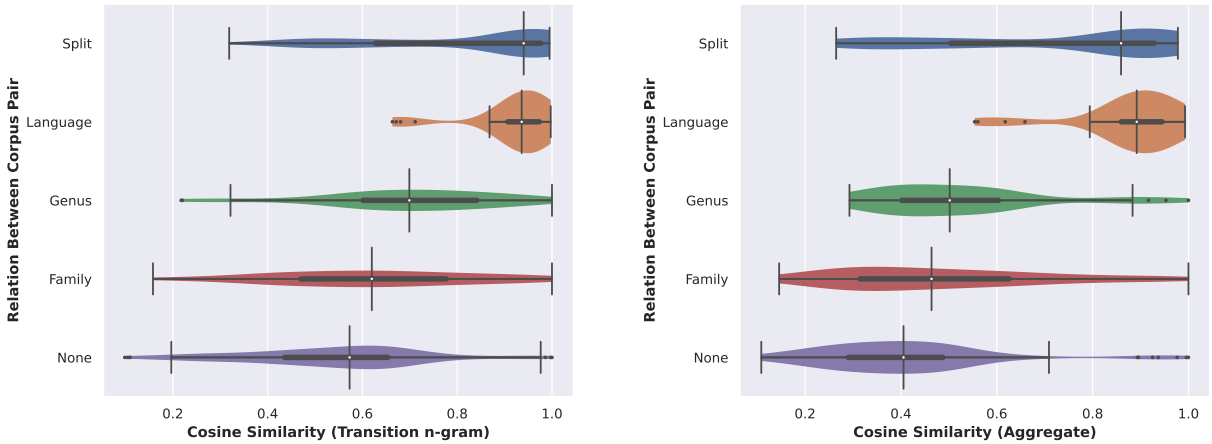


Figure 5.3: Violin plots, marked with quartiles, showing cosine similarities of transition n-gram distributions alone (left) as well as the average similarities across all n-gram distributions (right) across every pair of corpora, including training and test splits counted separately.

5.5 Model-Independent Demonstrations

Before testing out the metrics with the testbed models, I look at their patterns independent of the models. First, I visualize the n-gram cosine similarity metrics to determine whether they yield scores in proportion to how closely related the languages of the two corpora are.

To get the graphs in Figure 5.3, I compare all possible pairs from among the corpora in use (see Table 3.3), including training and testing splits separately to enable comparisons across a *split* relation category in addition to the language relatedness categories. Comparisons between training and testing splits of the same corpora and between corpora of the same language have a very high level of cosine similarity for transition n-gram distributions – close to the maximum of 1. The more distantly related the languages of the corpora, the lower the average cosine similarity. This is consistent with expectations of syntactic similarity among more closely related languages.

The aggregate similarity shows a weakened pattern: while corpora of the same language are still very similar, related language corpora are only slightly more similar to one another than are unrelated language corpora. Since the aggregate metric accounts for character and word n-grams, it makes sense that cross-lingual pairs are all dissimilar by this measure. Only corpora of the same language tend to share sequences of words, and related languages often differ in writing systems.

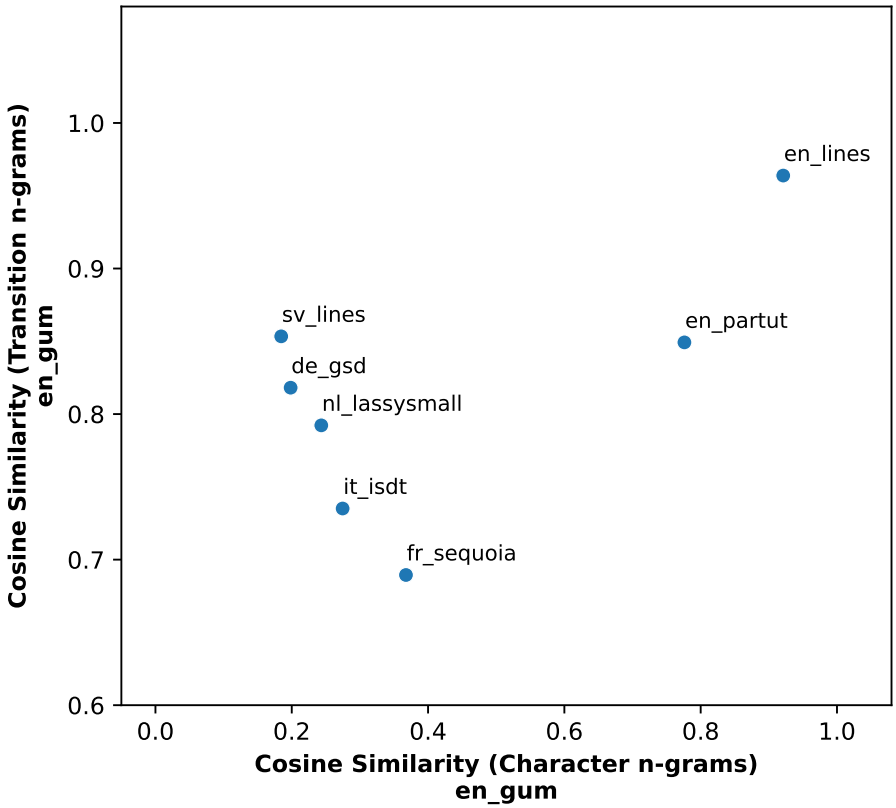
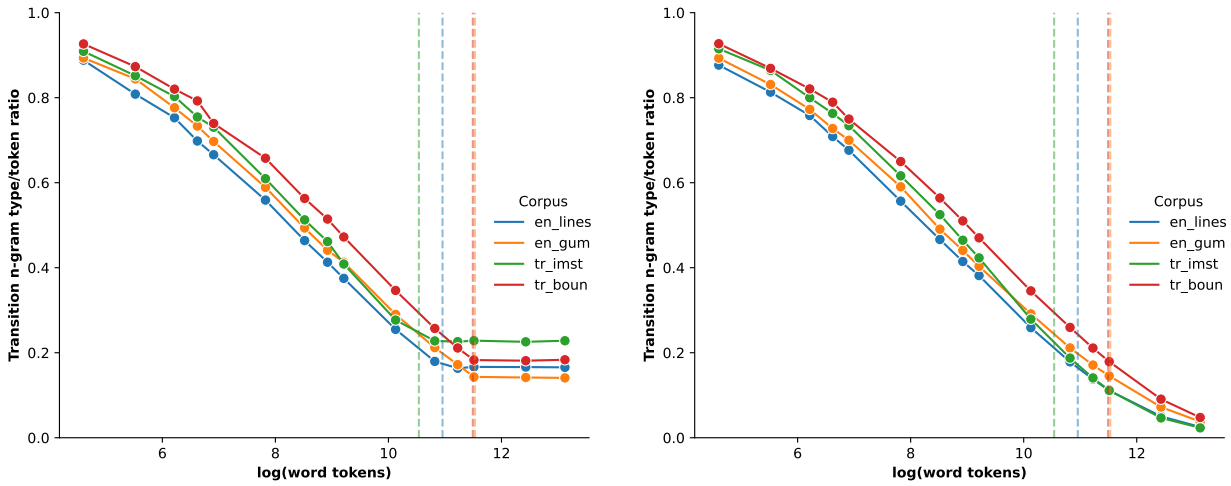


Figure 5.4: An assortment of Germanic and Romance language corpora compared to an English corpus from UD (Zeldes, 2017) based on two different similarity metrics. English appears to be most orthographically similar to French (x-axis) and syntactically similar to Swedish (y-axis).

To observe the different patterns captured by ostensibly-orthographic measures of character n-grams vs. ostensibly-syntactic measures of transition n-grams, I also compare the respective cosine similarities given an English corpus and some Germanic and Romance language corpora. Figure 5.4 suggests that, by these metrics, English is syntactically like its fellow Germanic languages but written like the more distantly related Romance languages.

Notably, the Swedish corpus is even more syntactically similar to the primary English corpus than is one of the other English corpora. This is possibly a matter of genre – that is, the Swedish corpus only includes data from genres also covered in the primary English corpus (i.e., *fiction*, *nonfiction*, and *spoken*). Meanwhile, the less-similar English corpus includes data from both genres with coverage (i.e., *news* and *wiki*) and genres with no coverage (i.e., *legal*; see Table A.6) in the primary corpus.



(a) Only unique utterances counted.

(b) Repeated utterances counted.

Figure 5.5: Transition n-gram TTR calculated on incremental subsamples of two English and two Turkish corpora. Dotted lines represent the full corpus sizes.

Lastly, I compare transition n-gram TTR for English and Turkish corpora across different sample sizes. This is to tease out several different aspects of the corpora: the languages (and their intrinsic complexities, as seen in Figure 5.2), the full corpus sizes, the sample sizes, and the effects of only counting unique utterances. TTR should gradually approach zero as the sample/corpus size increases, representing the fact that the number of types increases much more slowly as the tokens increase constantly.

Figure 5.5 confirms that only counting unique utterances/sentences means that sampling with replacement doesn't yield misleadingly low TTR – *misleading*, that is, in the sense that metrics ought to capture the actual information content of the data. Fully repeated data doesn't count as data augmentation in training, so metrics shouldn't treat it as such, either.

At smaller sample sizes, the two Turkish corpora consistently have higher TTR than the two English corpora. The larger English corpus has higher TTR than the smaller English corpus, and the same goes for the Turkish corpora. The true TTR, once all of the unique utterances have been counted, shows a slightly different pattern: the Turkish corpora still have higher TTR, but within the two languages, the larger ones now have lower TTR.

There is much that can be gleaned about the corpora and their intrinsic complexities from these observations across sample sizes. First, TTR seems to successfully capture the difference in Turkish and English Zipfian distributions, first noted in Figure 5.2.[1] Turkish, or at least the UD corpora representing Turkish, has relatively more uncommon occurrences than common ones, be they words or transition sequences. Second, the larger of each of the two language corpora seem to be intrinsically more complex than their smaller counterparts, but the effects of this are mitigated by the fact that they are larger corpora.

In all, the strong correlation of TTR to corpus size, plus the indication that it captures the complexity intrinsic to languages and corpora, is promising for the efficacy these metrics might have in measuring sparsity – or, more directly, predicting single-source performance better than corpus length/size. The consistency of the cosine similarities with known patterns of language relatedness is also promising with regard to the ability of the metrics to capture linguistically relevant information.

Even the initial comparison of English to its Germanic and Romance relatives already opens up some interesting questions about what data can indicate about the typological properties of English. Once the best predictive metrics have been selected, the analyses they enable should be similarly fruitful.

[1] Although the initial example is for a word-level vocabulary, the pattern extends to the n-gram vocabularies.

CHAPTER 6
Single-Source Baselines

6.1 Parsing Results

Of the 40 UD corpora listed in Table 3.3 in Chapter 3, I use 35 to train single-source parsing models. Table 6.1 offers a look at the performance of the models for corpora that were part of Zeman et al.'s (2018) shared task, both in terms of LAS and in terms of (hypothetical) scoreboard position, compared to the baseline and best systems and their true scoreboard positions. The complete list of UAS, LAS, and POS results for each of these is in Appendix A.

		LAS	
TARGET CORPUS	BASELINE CONLL	BEST CONLL	SINGLE-SOURCE
ar_padt	66.41 (20)	77.06 (1)	80.27 (1)
bxr_bdt	12.61 (10)	19.53 (1)	15.28 (7)
de_gsd	70.85 (18)	80.36 (1)	79.60 (2)
en_gum	74.20 (19)	85.05 (1)	82.42 (4)
en_lines	73.10 (18)	81.97 (1)	80.14 (4)
fa_seraji	79.10 (20)	88.11 (1)	82.95 (14)
fi_ftb	75.64 (18)	88.53 (1)	82.47 (13)
fr_sequoia	81.12 (20)	89.89 (1)	86.51 (10)
hsb_ufal	23.64 (16)	46.42 (1)	22.54 (21)
it_isdt	86.26 (19)	92.00 (1)	89.29 (12)
ja_gsd	72.32 (19)	83.11 (1)	91.93 (1)
kk_ktb	24.21 (3)	31.93 (1)	27.45 (2)
kmr_mg	23.92 (9)	30.41 (1)	23.08 (17)
ko_gsd	61.40 (20)	85.14 (1)	80.15 (12)
ko_kaist	70.25 (21)	86.91 (1)	84.19 (14)
nl_lassysmall	74.56 (21)	86.84 (1)	82.17 (7)
sv_lines	74.06 (19)	84.08 (1)	80.30 (10)
sv_talbanken	77.91 (17)	88.63 (1)	82.20 (13)
tr_imst	54.04 (20)	66.44 (1)	62.49 (8)
ug_udt	56.26 (17)	67.05 (1)	63.52 (4)
zh_gsd	57.91 (19)	76.77 (1)	73.04 (2)

Table 6.1: LAS of the single-source models vs. entries in Zeman et al. (2018). Ranks in parentheses are the true ranks of the baseline and best systems and the hypothetical ranks of the new models.

With 26 entries populating the scoreboards for each corpus, the newly trained single-source models hover around the middle to the top half of the rankings, consistently exceeding the baseline systems. Worth noting is that the new models have the advantage of using gold standard tokenization, which is the likely reason for the particularly strong perfomance on the Arabic and Japanese corpora. Also worth noting is that the top-performing systems use concatenative multi-source training (among other pretraining and data augmentation strategies), while the baseline systems do not, so the new single-source models should be expected to match more closely with the baselines.

The single-source models trained per Chapter 4 appear, overall, competitive with comparable existing models. Even with the caveats to the comparisons, the results are reasonable enough to go forward with metric analysis and multi-source experiments.

6.2 Metrics for Predicting Single-Source Performance

6.2.1 Experimental Setup

Per Chapter 5, I precede the multi-source experiments by testing predictive metrics for single-source performance. The primary goal of this is to model intrinsic corpus sparsity and complexity, which I expect to be a significant factor in multi-source improvement. Given a certain level of similarity between target and supplemental corpora, smaller target corpora are more likely to see improvement in multi-source conditions. Low-resource languages simply have more to gain from data augmentation, in general.

Corpus size, as measured by length in word tokens, is the most straightforward measure of sparsity. I propose, however, that other metrics accounting for distributional properties of a corpus can do better by also capturing, for example, higher complexity inherent to some languages. This could be in terms of morphology, word order, or other known linguistic factors; a number of the proposed metrics are theoretically better equipped than the word count to capture these phenomena.

To test this, I calculate intra-corpus metrics on the training corpora, in addition to some "cross-corpus" metrics that compare training and test splits. I then perform regression tests and rank the metrics by correlation with test set LAS. The full results are shown in Table 6.2.

	Metric Name	R^2	p-value
	Type-Token Ratios (Train Set)		
1	transition n-grams	0.959	1.8e-24
3	aggregate	0.945	2.3e-22
4	gov'ship n-grams	0.944	2.8e-22
7	POS n-grams	0.925	3.5e-20
9	rel n-grams	0.922	7.5e-20
22	POS/rel n-grams	0.862	1.0e-15
24	log(POS + arcs)	0.852	3.0e-15
32	words	0.803	3.4e-13
54	char n-gram + arcs	0.620	2.0e-08
62	char n-grams	0.550	3.4e-07
	Entropies (Train Set)		
42	log(char n-gram + arcs)	0.733	5.4e-11
48	log(char n-grams)	0.679	1.2e-09
49	log(POS/rel n-grams)	0.669	2.0e-09
51	log(aggregate)	0.649	5.4e-09
52	log(gov'ship n-grams)	0.643	7.0e-09
53	log(rel n-grams)	0.638	8.8e-09
61	log(words)	0.556	2.7e-07
63	log(POS n-grams)	0.531	7.0e-07
65	log(transition n-grams)	0.504	1.8e-06
77	POS + arcs	0.001	8.7e-01
	Cosine Similarities (Train vs. Test Set)		
11	POS/rel n-grams	0.920	1.2e-19
19	rel n-grams	0.881	8.0e-17
25	aggregate	0.842	8.6e-15
26	transition n-grams	0.840	1.1e-14
27	POS n-grams	0.838	1.4e-14
30	gov'ship n-grams	0.828	3.6e-14
45	words	0.708	2.4e-10
50	log(char n-gram + arcs)	0.668	2.1e-09
57	log(char n-grams)	0.604	4.0e-08
59	POS + arcs	0.601	4.6e-08
	Other Dependency Metrics (Train Set)		
66	nonplanar %	0.262	1.7e-03
68	swaps/utterance	0.212	5.4e-03
69	nonprojective %	0.208	5.9e-03
70	relative gov dist	0.180	1.1e-02
71	illnested %	0.169	1.4e-02
72	log(depth/word)	0.160	1.7e-02
73	log(neighbor dist)	0.156	1.9e-02
74	log(deps/gov)	0.111	5.0e-02
75	absolute gov dist	0.055	1.7e-01
76	log(avg gap degree)	0.042	2.4e-01

	Metric Name	R^2	p-value
	Left Mass Percentages (Train Set)		
2	transition n-grams	0.951	3.7e-23
5	aggregate	0.934	4.3e-21
6	gov'ship n-grams	0.932	8.3e-21
8	rel n-grams	0.924	5.2e-20
10	POS n-grams	0.920	1.2e-19
18	POS/rel n-grams	0.883	6.2e-17
21	POS + arcs	0.868	4.4e-16
31	log(words)	0.815	1.2e-13
46	log(char n-gram + arcs)	0.700	3.8e-10
55	log(char n-grams)	0.611	2.9e-08
	Type Counts (Train Set)		
33	log(gov'ship n-grams)	0.795	6.6e-13
34	log(aggregate)	0.784	1.6e-12
35	log(POS/rel n-grams)	0.782	1.9e-12
37	log(rel n-grams)	0.781	2.1e-12
38	log(POS n-grams)	0.772	4.0e-12
39	log(words)	0.768	5.2e-12
40	log(transition n-grams)	0.766	6.2e-12
41	log(char n-gram + arcs)	0.735	4.9e-11
44	log(char n-grams)	0.733	5.6e-11
47	log(POS + arcs)	0.686	8.4e-10
	Cross-Entropies (Train vs. Test Set)		
12	gov'ship n-grams	0.913	4.4e-19
13	rel n-grams	0.905	2.0e-18
14	aggregate	0.904	2.2e-18
15	transition n-grams	0.904	2.4e-18
16	log(POS/rel n-grams)	0.900	4.4e-18
17	words	0.885	4.9e-17
20	POS n-grams	0.876	1.5e-16
36	log(char n-gram + arcs)	0.781	2.0e-12
43	log(char n-grams)	0.733	5.6e-11
67	log(POS + arcs)	0.227	3.8e-03
	Other Surface Metrics (Train Set)		
23	log(word tokens)	0.855	2.2e-15
28	log(utterance tokens)	0.837	1.5e-14
29	log(char tokens)	0.834	2.0e-14
56	log(word n-grams / word)	0.604	4.0e-08
58	char n-grams / word	0.602	4.3e-08
60	chars / word	0.583	9.6e-08
64	words / utterance	0.517	1.2e-06

Table 6.2: Correlations of the various metrics to single-source LAS on the related test sets. Metrics are grouped by subtypes and marked with their overall ranking in terms of highest correlation.

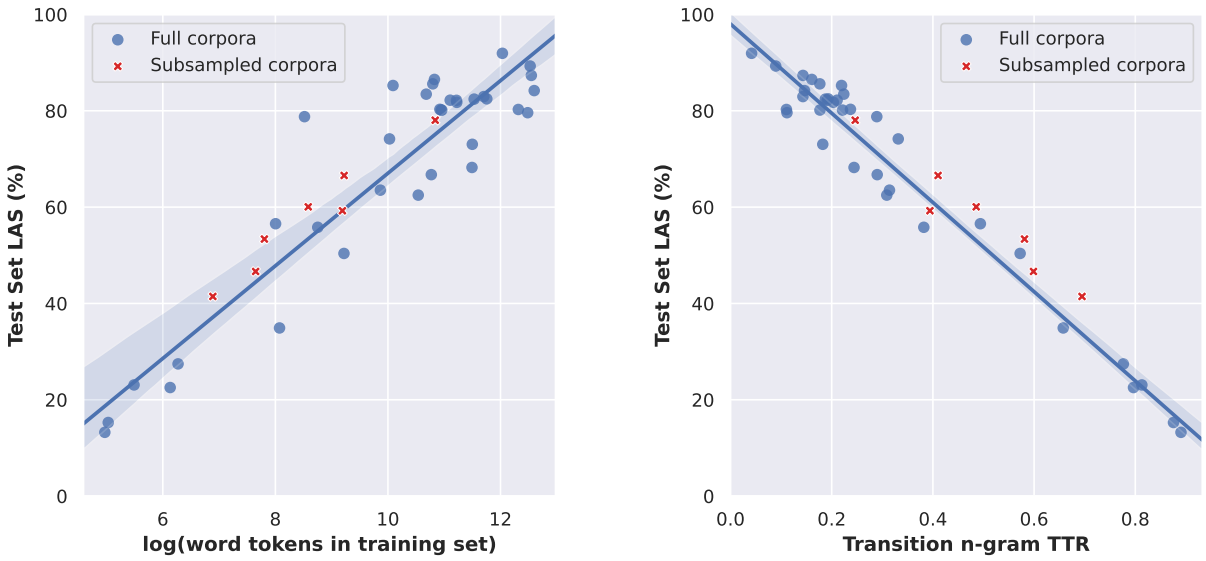


Figure 6.1: Regression lines showing the strength of logarithmic word tokens (left) and transition n-gram TTR (right) in predicting single-source LAS.

6.2.2 Testing & Selection

There are a number of takeaways from these tests. In summary:

- Transition n-gram TTR is the strongest predictor of single-source LAS.
- Corpus size as measured by word tokens has a logarithmic relationship with LAS and falls within the top quartile of metrics for predicting single-source LAS.
- TTR and the novel *left mass percentage* metrics perform similarly across all the feature sets, which suggests that they capture the same distributional information.
- "Cross-corpus" metrics between the train and test sets are strong LAS predictors but are still not as strong as TTR and left mass percentage metrics on the training sets alone.
- None of the non-projectivity and other dependency metrics are strong predictors of LAS, likely because they are not proportional to corpus size.

Figure 6.1 shows the correlations between LAS and the word token metric vs. transition n-gram TTR. The TTR metric is actually inversely related to LAS, but the correlation is near-perfectly linear.

The correlation graphs also include points representing models trained on *subsampled* corpora. These corpora, which include the English GUM corpus (Zeldes, 2017) as well as Maltese (Čéplö, 2018) and Uyghur (Aili et al., 2016) corpora, aren't used for the regression calculations, but the lines fit them well nonetheless. Subsampling becomes a more important point of analysis in Chapter 7, but here it helps to fill out the mid-level resource space that few UD corpora fall into, while showing that large-but-subsampled corpora act similarly to corpora that are actually of the smaller size. In other words, it's reasonable to assume that patterns observed in low-resource corpora are because of their smaller size (or higher TTR) and not some other confounding factor.

Regarding the use of training vs. test sets in the metrics, I also calculated the intra-corpus metrics on test sets alone in preliminary experiments. The results were unilaterally less predictive than the same metrics calculated for the training sets, however. While initially surpising, – since test set performance is what the metrics are trying to predict – this is consistent with LAS performance being beholden to the amount of training data. Any metric that ignores the amount or quality of training data is missing the most important factor, even if it manages to capture the intrinsic complexity of the language via the test set.

While the correlations are strong overall, it's worth reiterating that, since the data points used for calculating R^2 values are the same ones to which the regression models are fitted, the metrics are technically demonstrating *descriptive* more than *predictive* power. The variety of corpora and the simple regression mitigate the ill effects of overfitting, and the strength of fit for the better metrics seems likely to generalize, but it's still likely that held-out data would show greater deviation.

Ultimately, the n-gram TTR metrics are the strongest predictors of single-source LAS, and thereby the strongest indicators of the intrinsic sparsity and complexity of the target corpora. That transition n-gram TTR *specifically* is the strongest makes sense, given that transition sequences are what the models are learning to predict. Beyond that, the oracles approximate syntactic structure while also including the arc relation labels and POS tag labels; they effectively combine information from several of the other feature categories into one.

A regression model that reliably predicts LAS is interesting and useful on its own. For now, however, the main purpose of n-gram TTR is in aiding prediction of multi-source improvement.

CHAPTER 7
Multi-Source Experiments

7.1 Parsing Results

In total, I train parsers under 50 different multi-source conditions, covering 19 unique target corpora. Table 7.1 shows the breakdown of the multi-source training experiments in terms of the training set size and language family of the target corpus, as well as the relation between the languages of the target and supplemental corpora.

Indo-European languages and moderately large corpora form large contingents of the experiments, but the spread is reasonably representative of the makeup of UD. The count for the *No relation* category of language relatedness is somewhat inflated due to the classification of Turkish-German. UD labels *code-switching* as a language family, and while I could hand-label Turkish-German in order to represent its connection to Turkish and German, there isn't a systematic way to do so – and it would be the only language the change applied to.

Table 7.2 shows a comparison of LAS results for models with target corpora included in Zeman et al.'s (2018) shared task. Single-source results for the target corpora are also shown for reference. As with the single-source models, fully tabulated UAS, LAS, and POS results are in Appendix A.

MULTI-SOURCE EXPERIMENTS (VS. UNIQUE CORPORA)					
Target Corpus Size		***Target Language Family***		***Target-Supplement Language Relation***	
<1K tokens	18 (5)	IE	22 (9)		
1-10K tokens	5 (3)	Turkic	10 (4)	Same language	7
10-100K tokens	25 (10)	Code-switching	6 (1)	Same genus	13
>100K tokens	2 (1)	Afro-Asiatic	5 (1)	Same family	9
		Mongolic	3 (1)	No relation	21
		Uralic	2 (1)		
		Dravidian	2 (2)		
Total	50 (19)				

Table 7.1: Composition of the multi-source experiments, in three dimensions. The primary numbers count all experiments with a characteristic; numbers in parentheses count unique target corpora.

			LAS	
CORPUS NAME	BASELINE CONLL	BEST CONLL	SINGLE-SOURCE	BEST MULTI-SOURCE
bxr_bdt	12.61 (10)	19.53 (1)	15.28 (7)	18.78 (3)
en_gum	74.20 (19)	85.05 (1)	82.42 (4)	82.68 (4)
en_lines	73.10 (18)	81.97 (1)	80.14 (4)	82.82 (1)
hsb_ufal	23.64 (16)	46.42 (1)	22.54 (21)	46.54 (1)
kk_ktb	24.21 (3)	31.93 (1)	27.45 (2)	32.41 (1)
kmr_mg	23.92 (9)	30.41 (1)	23.08 (17)	29.78 (2)
sv_lines	74.06 (19)	84.08 (1)	80.30 (10)	82.63 (2)
sv_talbanken	77.91 (17)	88.63 (1)	82.20 (13)	83.93 (12)
tr_imst	54.04 (20)	66.44 (1)	62.49 (8)	66.35 (2)
ug_udt	56.26 (17)	67.05 (1)	63.52 (4)	63.69 (4)

Table 7.2: LAS of the best multi-source models for the target corpora with counterparts in Zeman et al. (2018), alongside the single-source models and shared task entries. Ranks in parentheses are the true ranks of the baseline and best systems and the hypothetical ranks of the new models.

For every target corpus, the best multi-source model outperforms the single-source model, if not always to the point of significance. Many of the models reach the top of the scoreboard, although some are unfair comparisons, using supplemental corpora that did not yet exist during the shared task (in addition to the caveat that the new models use gold standard tokenization).

Table 7.3 provides another look at the multi-source results – this time, showing the best LAS for each target corpus that has multiple multi-source training conditions in the suite of experiments. The supplemental corpora are specified here, as well. Turkish-German has the only condition that involves multiple supplemental corpora, which also happens to be its best condition.

Finally, Table 7.4 shows results of subsampling experiments for English, Maltese, and Uyghur. Each of these experiments gets a closer look in the case studies, but it's at least worth noting in the meantime that the Uyghur results in Table 7.3 show that even the best multi-source condition yields no significant improvement over the single-source model using the full training set, whereas the models trained with a subsample of Uyghur benefit from supplemental Turkish corpora.

Primary Corpus	Single-Source LAS	Supplemental Corpus	Multi-Source LAS	Predicted
bxr_bdt	15.28	kk_ktb	18.78*	×
en_gum	82.42	en_lines	82.68	✓
en_lines	80.14	en_gum	82.82*	✓
hsb_ufal	22.54	pl_pdb	46.54*	✓
kk_ktb	27.45	tr_boun	32.41*	✓
kmr_mg	23.08	fa_seraji	29.78*	✓
lt_hse	34.91	lt_alksnis	51.89*	✓
mt_mudt	74.16	it_isdt	74.97	✓
olo_kkpp	13.24	fi_ftb	30.44*	✓
qtd_sagt	50.38	de_gsd, tr_boun	58.86*	✓
sv_lines	80.30	sv_talbanken	82.63*	✓
ug_udt	63.52	tr_boun	63.59	✓

Table 7.3: The results of the best multi-source training conditions for each target corpus with more than one such condition trained. Starred results indicate significant improvement. Predictions represent whether the final predictive methodology correctly determined the best condition for the target corpus from among the experiments.

Target Corpus	Size	Single-Source LAS	Supplemental Corpus	Multi-Source LAS	*Significance* Prediction	*Significance* True
en_gum	5K (5%)	60.06	en_lines	73.83	✓	✓
			sv_lines	63.39	✓	✓
en_gum	10K (10%)	66.57	nl_lassysmall	67.93	✓	✓
			en_lines	80.36	✓	✓
			fr_sequoia	69.28	✓	✓
			sv_lines	69.47	✓	✓
			tr_boun	66.22	×	×
en_gum	51K (50%)	78.04	en_lines	80.52	✓	✓
			sv_lines	79.34	×	×
mt_mudt	2K (10%)	53.37	it_isdt	55.01	✓	✓
ug_udt	981 (5%)	41.45	tr_boun	44.27	✓	✓
ug_udt	2K (10%)	46.64	fa_seraji	44.23	×	×
			tr_boun	49.56	✓	✓
ug_udt	9K (50%)	59.25	tr_boun	60.76	✓	✓

Table 7.4: The results of subsampled multi-source training. Predictions are those of the final predictive methodology, regarding whether a given condition would yield significant improvement; the *True* column represents the actual significance of the improvement.

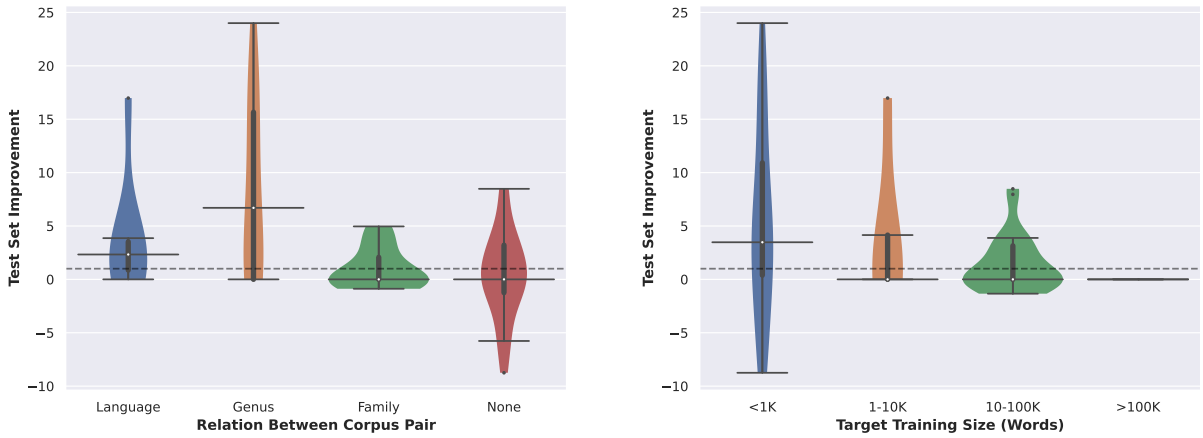


Figure 7.1: Violin plots showing the distributions of multi-source improvement for categories of target-supplement language relatedness (left) and target corpus size (right). Dotted lines represent the threshold of significance.

7.2 Metrics for Predicting Multi-Source Improvement

7.2.1 Experimental Setup

As discussed in Chapter 5, the main purpose of training such a wide array of multi-source models is to evaluate the predictive power of the metrics. Can the metrics predict whether a multi-source condition will yield significant improvement over a single-source model for the target corpus?

Improvement is measured by comparing multi-source LAS to the corresponding single-source LAS: it's equal to the difference between the two when that difference is significant per a bootstrap test, and equal to zero otherwise (see Equation 5.1). The job of the metrics, in order to make the binary significance prediction, is to recreate the significance threshold (chosen to be 1%).

I begin with the hypothesis that improvement is a function of *both* the similarity (or compatibility) between the target and supplemental corpora *and* the intrinsic sparsity (and/or complexity) of the target corpus. Figure 7.1 shows the distributions of improvement for each of the multi-source models, categorized in terms of language relatedness and target corpus size, as in Table 7.1. Consistent with my hypothesis, the average improvement is significant when the target and supplemental corpora are of the same language or genus, as well as when the target corpus is very small. The next step, then, is to turn these patterns into predictions.

The metrics used to make predictions about single-source LAS in Chapter 6 need to be modified to be applicable to multi-source improvement. Cross-corpus metrics like cosine similarity and cross-entropy now compare the target and supplemental training corpora, for one.[1]

Given the efficacy of TTR metrics in the single-source experiments, their inclusion in the multi-source experiments is a given – with some adjustments. TTR calculated on the concatenated corpora, for example, isn't meaningful, because the concatenation has different composition from the target test set. It would be a different matter if the test sets also included the supplemental corpora, but they do not.

Instead, I still calculate TTR for the target training set on its own. This first serves as a baseline: how predictive is target corpus sparsity alone, before even considering the supplemental corpora? It also factors into two types of composite metrics: cosine similarities and cross-entropies between the target and supplemental corpora, scaled by the target corpus TTR. These composite metrics test my hypothesis regarding the twin factors of similarity and sparsity.

I include a few additional baseline metrics, as well:

- Target, supplemental, & total concatenated training corpus size
- Ratio of target to supplemental training corpus size
- Boolean functions of language relatedness (same language, genus, family, or script)
- "Sameness" as sum of boolean functions of language relatedness
- All significant improvement ("majority" baseline; maximizing baseline F1 score)

As discussed in Chapter 5, using correlation for prediction and R^2 for ranking doesn't make sense with the multi-source task formulation or expected distributions, unlike in the single-source experiments. Although being able to rank supplemental corpora is an important corollary, the main prediction is a binary one: is there significant improvement or not?

Given the improvement and metrics for each of the multi-source conditions, I find the best thresholds given each metric for separating the significantly improved models from the rest. These best thresholds serve as the predictive model for each metric, which I then rank by F1 scores.

[1] As mentioned in the first round of predictive metric experiments, I did some preliminary testing using the target test set rather than the target training set for intra-corpus metrics, but these were unilaterally worse for prediction.

	Metric Name	F1	Acc	p-value
	CS * TTR			
1	+aggregate	93.62	94.00	5.6e-06
2	+POS n-grams	90.20	90.00	5.7e-06
3	+transition n-grams	83.02	82.00	2.2e-04
7	+words	78.69	74.00	2.4e-04
10	+POS + arcs	78.43	78.00	2.6e-03
27	+char n-grams	71.43	76.00	2.0e-02
36	+rel n-grams	68.75	60.00	1.1e-01
38	+POS/rel n-grams	67.69	58.00	1.8e-01
39	+gov'ship n-grams	67.69	58.00	1.8e-01
45	+char n-gram + arcs	66.67	70.00	6.1e-02
	Cosine Similarities (Target vs. Supplemental)			
5	+words	80.00	80.00	1.5e-03
25	+POS + arcs	72.73	64.00	7.8e-03
26	-POS/rel n-grams	71.70	70.00	2.7e-02
33	+POS n-grams	69.57	58.00	6.2e-02
34	-gov'ship n-grams	69.23	68.00	5.2e-02
35	-rel n-grams	69.23	68.00	5.2e-02
42	+aggregate	66.67	56.00	2.9e-01
44	+transition n-grams	66.67	58.00	2.3e-01
47	+char n-grams	65.63	56.00	3.4e-01
49	+char n-gram + arcs	64.79	50.00	1.0e+00
	Type-Token Ratios (Target)			
8	+aggregate	78.43	78.00	2.6e-03
9	+char n-gram + arcs	78.43	78.00	2.6e-03
12	+POS + arcs	78.43	78.00	2.6e-03
13	+transition n-grams	78.43	78.00	2.6e-03
14	+POS/rel n-grams	78.43	78.00	2.6e-03
15	+gov'ship n-grams	78.43	78.00	2.6e-03
16	+rel n-grams	78.43	78.00	2.6e-03
17	+POS n-grams	78.43	78.00	2.6e-03
18	+char n-grams	76.92	76.00	4.3e-03
28	+words	71.19	66.00	3.5e-02

	Metric Name	F1	Acc	p-value
	CE * TTR			
4	+POS/rel n-grams	81.82	84.00	1.4e-03
6	+aggregate	80.00	78.00	7.3e-04
11	+POS + arcs	78.43	78.00	2.6e-03
19	+words	76.92	76.00	4.3e-03
21	+char n-gram + arcs	75.00	80.00	9.0e-03
22	+POS n-grams	74.51	74.00	1.1e-02
23	+transition n-grams	74.42	78.00	1.1e-02
24	+rel n-grams	74.42	78.00	1.1e-02
29	+char n-grams	70.83	72.00	2.9e-02
32	+gov'ship n-grams	70.00	64.00	5.7e-02
	Cross-Entropies (Target vs. Supplemental)			
30	+words	70.59	70.00	3.5e-02
37	-POS + arcs	68.57	56.00	1.2e-01
40	+char n-gram + arcs	67.65	56.00	2.2e-01
41	+char n-grams	67.65	56.00	2.2e-01
43	+aggregate	66.67	60.00	1.8e-01
46	+POS/rel n-grams	66.67	64.00	1.2e-01
51	+rel n-grams	63.49	54.00	5.5e-01
53	+transition n-grams	63.01	46.00	1.0e+00
54	+gov'ship n-grams	63.01	46.00	1.0e+00
55	+POS n-grams	63.01	46.00	1.0e+00
	Other Metrics			
20	-target size	75.47	74.00	7.2e-03
31	+size ratio	70.18	66.00	4.9e-02
48	+all significant	64.86	48.00	1.0e+00
50	-full-train-size	63.89	48.00	1.0e+00
52	+supplemental size	63.01	46.00	1.0e+00
56	+same family	60.38	58.00	3.8e-01
57	+sameness	59.57	62.00	2.5e-01
58	+same writing	59.37	48.00	1.0e+00
59	+same genus	59.09	64.00	2.0e-01
60	-same language	56.72	42.00	4.5e-01

Table 7.5: The best F1 scores using the various metrics to predict significant multi-source LAS improvement on the related test sets. Metrics are grouped by subtypes and marked with their overall ranking in terms of highest F1 scores; p-values come from McNemar's test (McNemar, 1947) in comparison to the *all significant* baseline. Signs (+/-) indicate the direction of the thresholds.

7.2.2 Testing & Selection

Table 7.5 shows the full results and rankings, including p-values calculated via McNemar's test (McNemar, 1947), which compares all of the models' contingency tables to that of the *all significant* baseline and represents whether they differ significantly from said baseline.

The rankings indicate that the TTR baseline is strong, as well: just knowing that the target training corpus is sparse (or not) is predictive of whether any multi-source conditions will yield improvement. This is likely an artefact of the experimental suite, which includes many small target corpora for which most of their supplemental corpora prove beneficial.

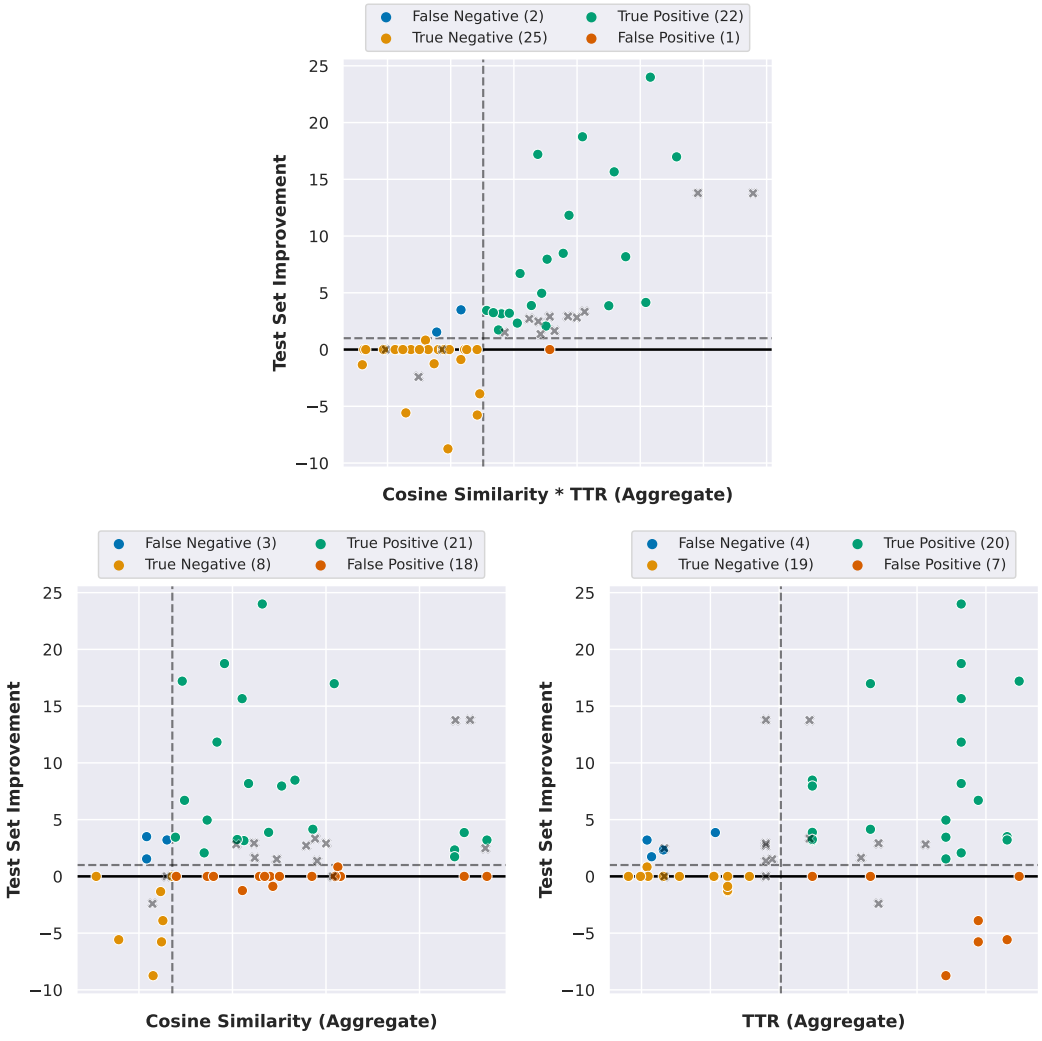


Figure 7.2: True vs. predictive thresholds for the best metric and its components.

Few of the purely cross-corpus metrics beat the TTR baselines, but the composite metrics are another story: the strongest predictor is the *aggregate* (i.e., average across features) *n-gram cosine similarity* scaled by *aggregate n-gram TTR*.[2] Figure 7.2 shows the true vs. predicted thresholds for this best metric and its two component metrics. These graphs visualize the tranformation of the improvement calculations and metrics for each condition into contingency tables.

Results for conditions with subsampled target corpora (see Table 7.4) are marked as gray X's. These aren't included in threshold calculation or evaluation, but they follow the same patterns as the full-size corpora, which strengthens their case as stand-ins for truly low-resource corpora.

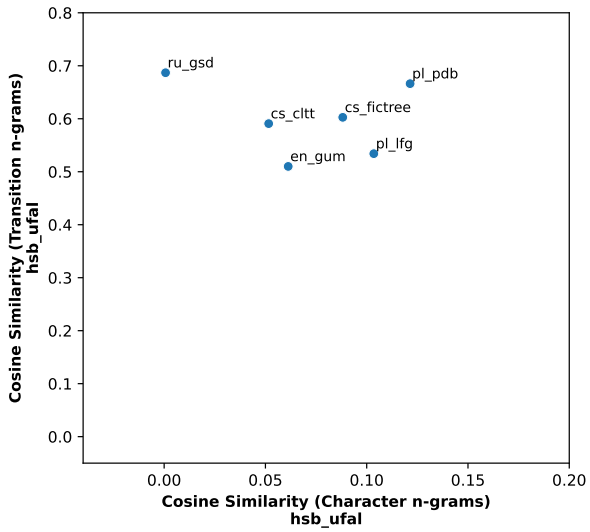
The cosine similarity graph makes it clear what a poor predictor similarity is on its own. Points on the far right edge of the graph are those where the target and supplemental corpora are of the same language; the fact that the biggest improvements here are for *subsampled* conditions shows the importance of the size/sparsity of the target training corpus.

The TTR graph, meanwhile, has an interesting shape; smaller target corpora (on the right) have greater capacity to improve than do larger target corpora (on the left). Models with sparse/low-resource/high-TTR target corpora are more likely to improve than not to. This is partially a result of my experimental design choices, and partially a result of it being genuinely difficult to select supplemental corpora that *won't* yield some improvement for especially small target corpora.

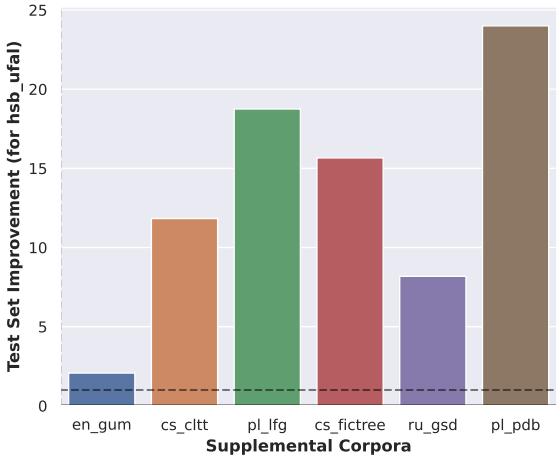
In spite of the insufficiency of the constituent metrics on their own, the composite metric (which simply multiplies the similarity and TTR) yields very strong F1 and accuracy scores – again, with the caveat that these scores are calculated for the same data points used to create the thresholds. The few cases it gets wrong are in (a) predicting that a larger Turkish corpus will be improved by a slightly smaller Turkish supplement, and (b) predicting that Buryat and Kazakh (both low-resource language corpora; Mongolic and Turkic, respectively) won't significantly improve one another.

Only the *TTR-scaled POS n-gram similarity* comes close to its aggregate counterpart in terms of F1 score. Since there is a clear winner among the metrics, the case studies that follow focus on the implications of using this predictive threshold – for *TTR-scaled aggregate n-gram cosine similarity* – for specific target corpora. (Tables 7.3 and 7.4 also show predictions using this metric.)

[2]It's surprising that cross-entropy underperforms cosine similarity even when also scaled by TTR, but as I mention in Chapter 5, this is likely because both use frequency-weighted feature distributions rather than conditioned sequences.



(a) Measures of orthographic and syntactic similarity for the Upper Sorbian corpus.



(b) True multi-source improvement, in ascending order of *predicted* improvement. The horizontal line is the true significance threshold. All models are predicted to yield improvement.

Figure 7.3: Metrics and results for Upper Sorbian.

7.3 Case Studies

7.3.1 Upper Sorbian's Familial Ties

Upper Sorbian is in an unusual position, as low-resource languages go. Its own UD corpus is tiny, while many of its neighbors in the Slavic branch of Indo-European have large UD corpora. With a high degree of intelligibility among Slavic languages in general, it seems perfectly positioned to benefit from concatenative training with these high-resource relatives.

Figure 7.3 juxtaposes similarity metrics between Upper Sorbian and other language corpora with the results of corresponding multi-source training conditions. The metrics, on their own, show that none of the other corpora have high character n-gram similarity with the Latin-script-using Upper Sorbian – even the ones that also use Latin scripts (e.g., Polish) as opposed to Cyrillic (Russian). They do show high syntactic similarity, including, surprisingly, for the more distantly related English. The results on the right show that the final predictions match up with the components on the left, for better and worse: all corpora are predicted and shown to yield significant improvement, but Russian is inaccurately predicted to be the second-best condition.

The binary significance predictions are correct, – even the edge case of English – while the actual order is somewhat mixed. The order predicted by the metric manages to correctly determine that the Polish *pl_ptb* corpus yields the best multi-source Upper Sorbian condition, but it greatly overestimates the efficacy of the Russian corpus.

In all, the predictions are right in the most important ways, but the metric is biased toward conditions where the supplemental corpora have high syntactic similarity with the target corpus – underplaying the degree to which the disjoint writing systems used by Upper Sorbian and Russian dampen the possible improvement. More sophisticated weighting of the component similarity metrics in the aggregate metric would likely lead to better predictions.

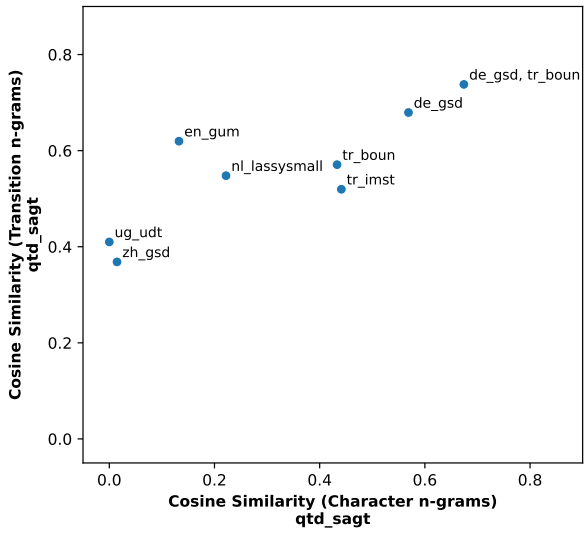
7.3.2 The Best of Both Worlds with Code-Switching

While a number of corpora in UD are for languages with extensive contact histories, Turkish-German is, as of writing, the only complete code-switching corpus that comes with both a training set and lexical data. It also happens to involve two languages with large UD corpora of their own.

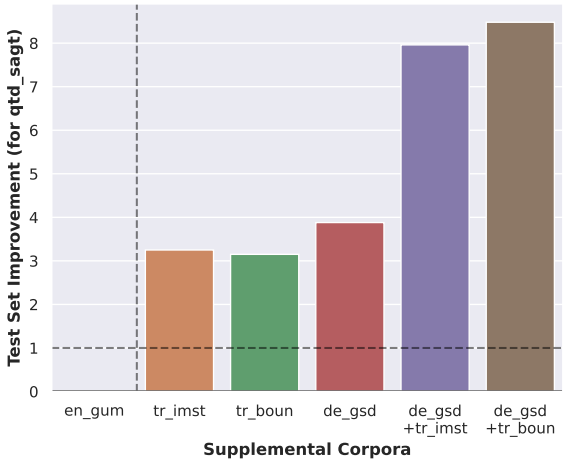
Harkening back to the first example in this dissertation, Example 1, Turkish-German code-switching looks like a straightforward combination of Turkish and German. It seems like it should benefit not just from Turkish and German supplemental corpora individually, but from both corpora simultaneously.

Figure 7.4 shows a pretty picture-perfect application of the predictive metric. The individual cosine similarity metrics on the left show that Turkish and German both get high degrees of similarity, and the concatenation of a Turkish and a German corpus is even more similar to Turkish-German than the sum of its parts – all unsurprising. The English corpus gets a high degree of syntactic similarity reminiscent of what happened with Upper Sorbian.

The actual predictive metric, however, is completely successful in predicting Turkish-German improvement across the multi-source conditions. It correctly determines the best concatenation of Turkish and German (from among the two Turkish corpora), and it correctly determines that English will not yield significant improvement.



(a) Measures of orthographic and syntactic similarity for the Turkish-German code-switching corpus.



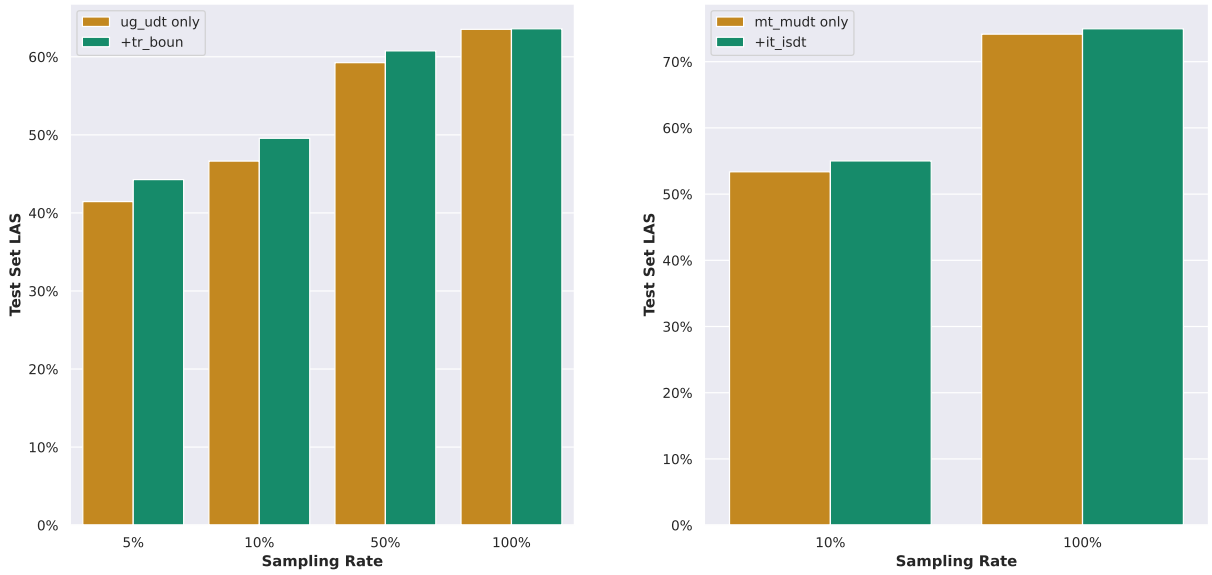
(b) True multi-source improvement, in ascending order of *predicted* improvement. The horizontal line is the true significance threshold; bars right of the vertical line are predicted to be significant.

Figure 7.4: Metrics and results for Turkish-German code-switching.

Most of these findings are in keeping with linguistic intuition. It doesn't take a sophisticated metric to decide that combining Turkish-German code-switching with Turkish and German corpora is probably a good idea. The ability of the predictive metric to capture the obvious cases with such accuracy, however, makes it all the more reasonable to rely upon it for the less-obvious cases, too – that is, for the target corpora for which the researcher may not have strong *a priori* knowledge. While the predictions aren't perfect, as seen with Upper Sorbian, the binary significance decisions and best condition selection are highly accurate. They also remain effective beyond single supplemental corpora, potentially enabling predictions for arbitrarily-many concatenated corpora.

7.3.3 Middling Resources, Middling Improvement

Upper Sorbian and Turkish-German code-switching are cases with small UD training corpora (under 10K tokens). While the Uyghur and Maltese corpora aren't *as* small, they're still sparse (around 20K tokens). Both represent languages with no particularly similar relatives: Maltese is a variety of Arabic with an Italic-influenced Latin script; Uyghur is one of the only Turkic languages to still use an Arabic script. As such, despite their size, they aren't likely to benefit from multi-source training.



(a) Single- vs. multi-source results for Uyghur.

(b) Single- vs. multi-source results for Maltese.

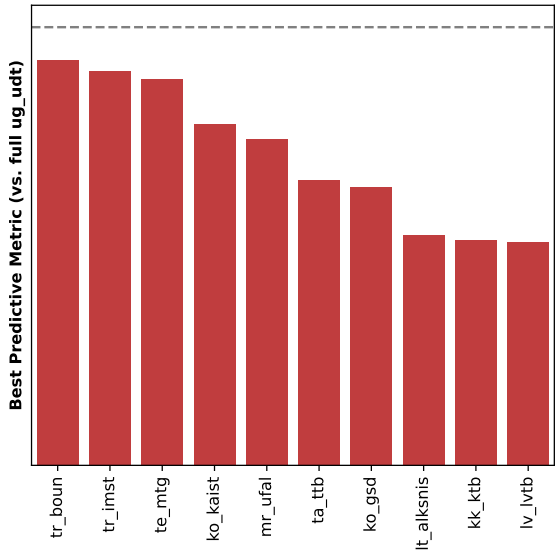
Figure 7.5: Results for Uyghur and Maltese with and without sampling.

Figure 7.6 shows that neither Uyghur nor Maltese is predicted by the metric to benefit from any multi-source training – given the *full* training sets. At 10% samples, however, they *are* predicted to benefit, as the predictive metric is sensitive to target corpus sparsity. The predictions bear out, and Figure 7.5 shows the relative results across sampling sizes for the best two supplemental corpora.

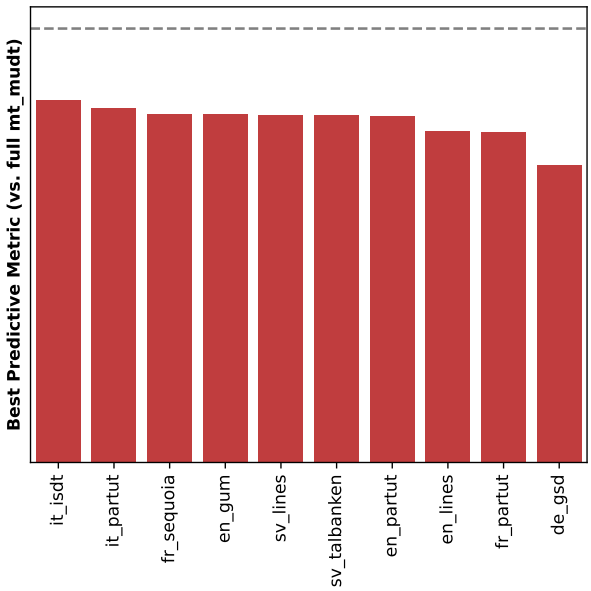
At a 50% sampling rate on the Uyghur training set, there is room for Turkish to improve upon the single-source baseline. At 5% and 10% sampling rates, the relative improvement of the Turkish multi-source conditions are greater – significant, if not exactly impressive. A moderate version of this is seen in Maltese, trained alongside an Italian corpus.[3] Although the improvement is still only modest for a 10% sample of the Maltese training set, it's enough to reach significance.

These subsampling experiments show that the predictive metric's sensitivity to target corpus sparsity is consistent with multi-source improvement. They also indicate a point at which improvement crosses from insignificance into significance. For corpora like Uyghur and Maltese, which are fairly different from their nearest neighbors, this point comes at a relatively small corpus size.

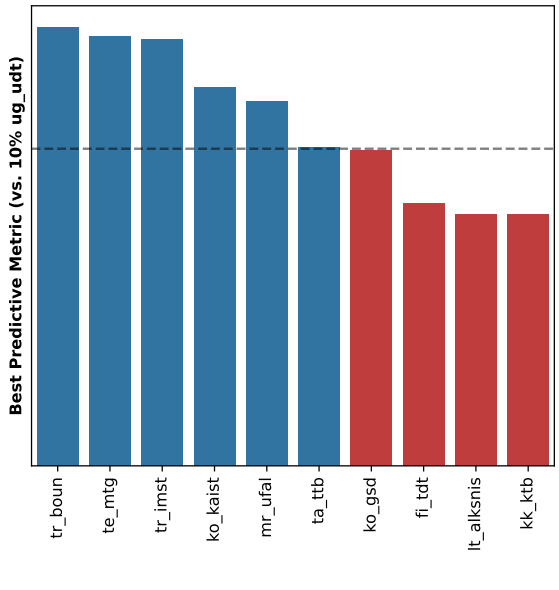
[3]The Arabic corpus, possibly because it's Modern Standard Arabic, comes nowhere close to promising improvement. Prior linguistic knowledge may or may not have helped in this case, making the metric especially helpful.



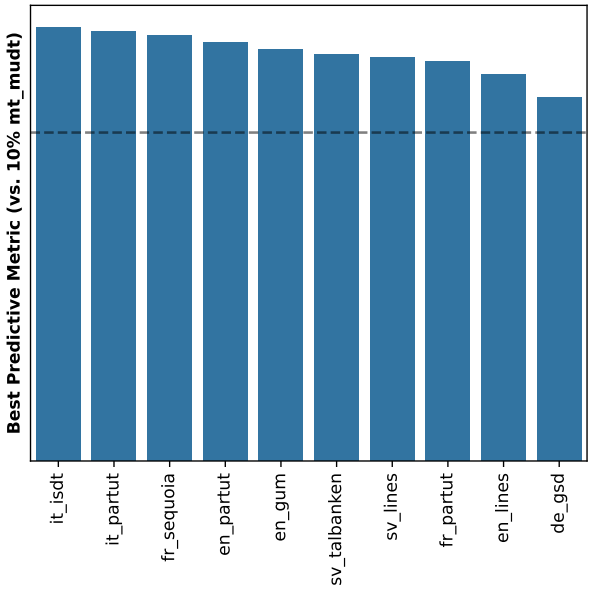
(a) Predictions for the full Uyghur corpus.



(b) Predictions for the full Maltese corpus.

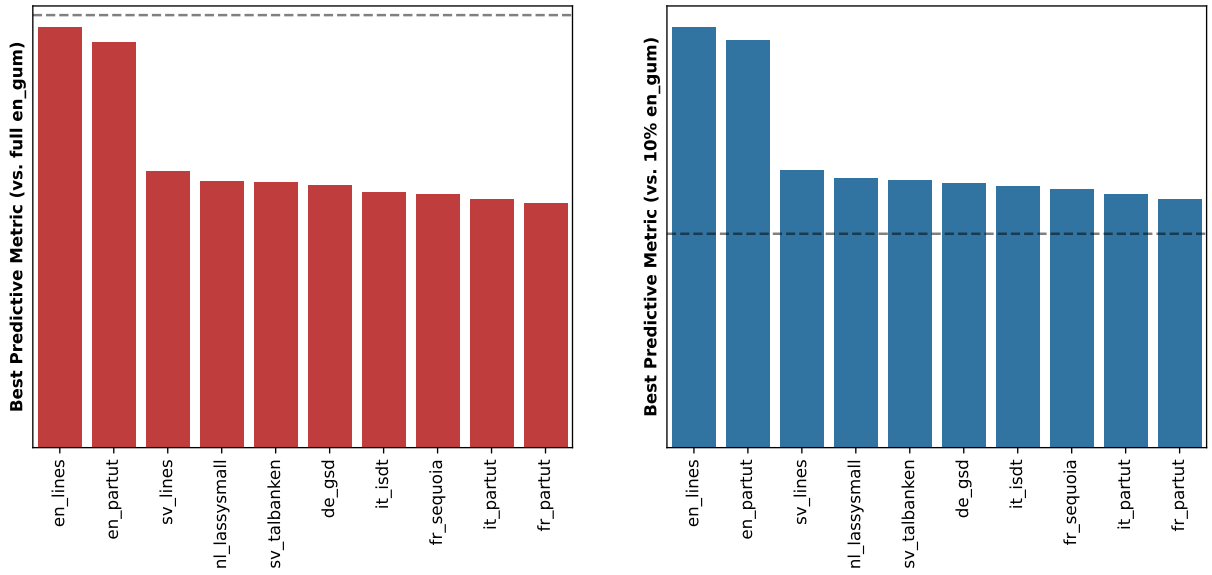


(c) Predictions for a 10% Uyghur sample.



(d) Predictions for a 10% Maltese sample.

Figure 7.6: The top ten conditions in terms of predicted multi-source improvement for Uyghur (Aili et al., 2016) and Maltese (Čéplö, 2018). Dotted lines represent the predictive threshold of significance. Blue bars indicate corpora that exceed the threshold, while red ones indicate corpora that fail to meet it.



(a) Predictions for the full corpus.

(b) Predictions for a 10% corpus sample.

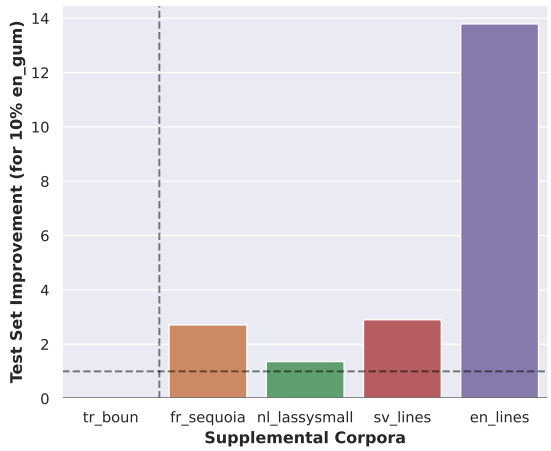
Figure 7.7: The top ten corpora in terms of predicted multi-source improvement for the English GUM corpus (Zeldes, 2017). Dotted lines represent the predictive threshold of significance. Blue bars indicate corpora that exceed the threshold, while red ones indicate corpora that fail to meet it.

7.3.4 What if English were a Low-Resource Language?

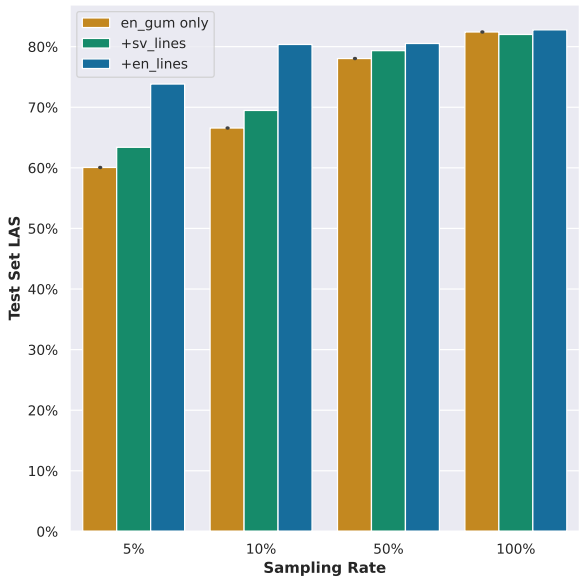
English is, if not the most-represented language in UD, inarguably one of the most-represented languages in NLP as a whole. With several large English corpora in UD, it's already unlikely enough that combining them yields much improvement for any one of their test sets – let alone that multi-source training with corpora of other languages should do anything for English. English is set up to be the ultimate supplemental language.

Historically, however, English has an interesting record of contact within the branches of the Germanic languages *and* with Romance languages. What, then, does the predictive metric have to say about how an English corpus would interact with these neighboring corpora if it weren't so large?

Figure 7.7 confirms that, at full size, even other English corpora are just barely predicted to be significant sources of improvement. At a 10% training sample, however, the world of multi-source training opens up, and all of the Germanic and Romance language corpora become viable.



(a) True multi-source improvement on a 10% corpus sample, in ascending order of *predicted* improvement. The horizontal line is the true significance threshold; bars right of the vertical line are predicted to be significant.



(b) Single- vs. multi-source results.

Figure 7.8: A visual comparison of models trained on the English GUM corpus (Zeldes, 2017), where the training set in question is subsampled (or not) at various rates.

Figure 7.8 shows how these predictions pan out. The multi-source models trained with a 10% sample for the English training set yield significant improvement for all the languages that are predicted as such. A different English corpus is far and away the best option, unsurprisingly; a Turkish corpus is predicted to be of little help, and indeed is not; and French, Dutch, and Swedish are all modestly helpful as supplemental corpora. The French corpus yields stronger improvement than the Dutch, against prediction. (Per Figure 5.4, its orthographic similarity is the likely reason.)

The Swedish corpus emerges as the best foreign language corpus to help train the English corpus. The graph on the right compares the multi-source conditions using Swedish and the other English corpus as supplements across target corpus sampling rates. The relative improvement of the same-language corpus increases greatly as the target corpus gets smaller, while the relative improvement from Swedish increases much more gradually. Although English won't be the primary beneficiary of multi-source training any time soon, the ability of the predictive metrics alongside subsampling tests to enable this kind of analysis could have implications for other languages, too.

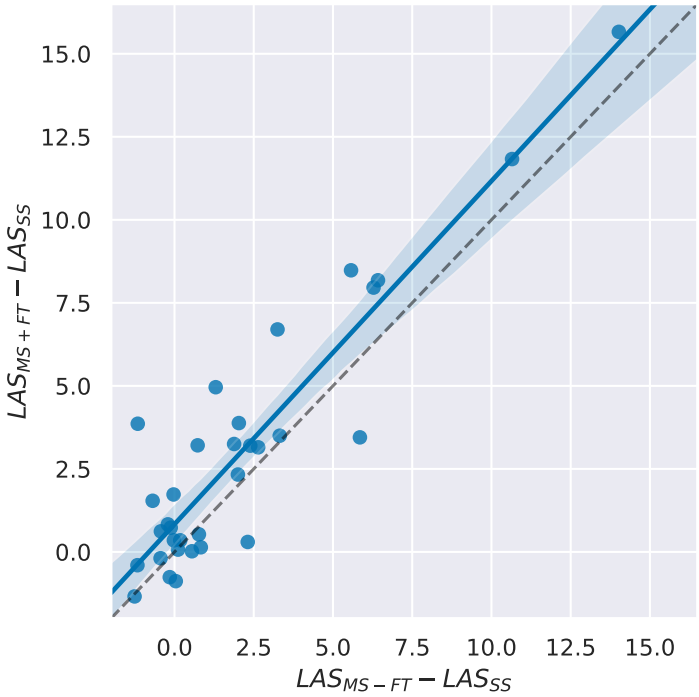


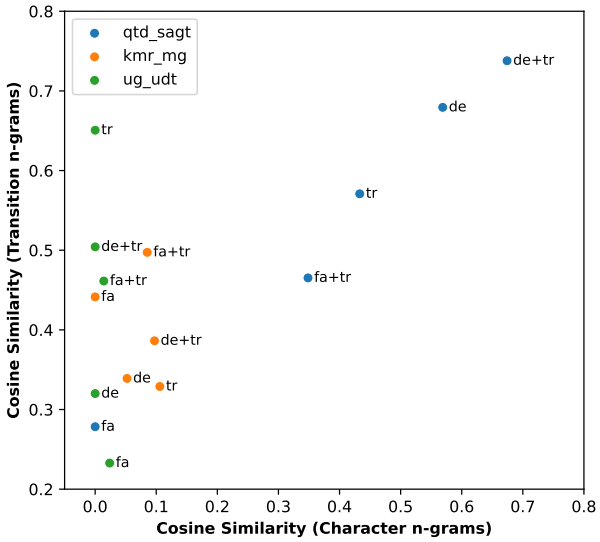
Figure 7.9: A comparison of multi-source training with and without fine-tuning, where each point represents a single condition (i.e., target and supplemental corpus). The diagonal dotted line has the slope of a one-to-one regression. The true regression line, in comparison, shows that the fine-tuned variants are generally better.

7.4 Peripheral Experiments

A few additional small-scale experiments, just outside the scope of the primary experiments, are worth discussing briefly. First, the issue of fine-tuning: as discussed in Chapter 4, all multi-source models are fine-tuned – the supplemental corpus is omitted from the last three training epochs.

Figure 7.9 confirms that training with or without fine-tuning is largely comparable, but fine-tuning has a slight edge. For the sake of simplicity, I deemed it worth sticking to fine-tuned results to maximize the chance of seeing significant differences with shorter training times.

Multi-source training without fine-tuning would make more sense in the case of multiple target corpora – that is, if the target vs. supplemental distinction were dissolved and performance on the latter mattered, as well. This would likely involve some focus on the balance of sizes among the various training corpora.



Trained Model	*Zero-shot LAS* kmr_mg	qtd_sagt	ug_udt
de_gsd	5.73	27.29	0.84
fa_seraji	**9.83**	3.02	7.55
tr_boun	8.86	19.57	**22.75**
fa_seraji, tr_boun	8.50	19.39	8.13
de_gsd, tr_boun	9.54	**45.84**	19.80

Table 7.6: Metrics and results for several Turkish-adjacent zero-shot experiments. Metrics are calculated based on the test splits for the testing languages, unlike previous calculations which used only training splits.

Although the results of my experiments can only speak to cases in which there is some small amount of in-domain training data for the target test set, I also look at a small number of zero-shot results. Table 7.6 looks at how well models trained on German, Persian, and Turkish perform on unseen data from Kurmanji, Turkish-German code-switching, and Uyghur – with some of the cosine similarity metrics for each of the training/testing language pairs included for reference.

Kurmanji, which has a very small training set and benefits greatly from multi-source training with Persian, does only slightly better with Persian than with the other languages in the zero-shot case due to their differing writing systems. Turkish-German code-switching, on the other hand, still benefits greatly from the concatenation of Turkish and German corpora, without the model needing to have been trained on code-switching specifically.

Uyghur, in contrast to Kurmanji, does well with the Turkish parser in spite of the disjoint writing systems. At first, this seems unintuitive, but the metrics on the left may explain it somewhat: Uyghur is more similar to Turkish than Kurmanji is to Persian, syntactically. The metrics still have explanatory power even in zero-shot modeling, so it seems they'd be worth adapting to that use case in future work.

Trained Model	*Zero-shot LAS* en_gum
en_lines	69.74
sv_lines	18.01
fr_partut	11.28
nl_lassysmall	10.15
tr_boun	5.53

Table 7.7: Results for zero-shot modeling of the English GUM corpus (Zeldes, 2017).

Table 7.7 shows similar zero-shot experiments for English, complementary to the metrics from Figure 5.4 and multi-source experiments from Figure 7.8. The parser trained on another corpus of the same language does quite well, and Swedish again comes in a distant but still notable second. The Dutch and French models have very similar performance, representing a relatively worse showing for French than in the multi-source case.

Building off of the prior set of zero-shot experiments, it seems like syntactic similarity is particularly important – hence a Turkish parser performing better on Uyghur data than a Persian parser performs on Kurmanji, and Swedish performing even better (and French slightly worse) relative to the other languages in parsing English from the multi-source to the zero-shot case. Again, the explanatory power of the metrics still holds promise in zero-shot modeling, but it would take another round of proper experiments to turn them into decent predictors.

These experiments also go to show, however, how useful even a small amount of in-domain data can be. The training corpus for Kurmanji is miniscule: only 242 tokens in total. Yet this is enough to bridge the gap between the target test set and the Persian corpus, allowing the Persian supplement to add around 4% improvement from single-source to multi-source LAS, despite the disjoing writing systems. Without those few hundred in-domain data, Persian is of little help in parsing Kurmanji. It's reassuring what even a small amount of in-domain data can accomplish.

Part III

Resolution

CHAPTER 8
Discussion

8.1 Are N-Gram Type/Token Ratios a Silver Bullet?

In both the single-source performance prediction and multi-source improvement prediction tests, TTR calculated on n-gram vocabularies, especially transitions and POS labels, proves incredibly effective. In regression from transition n-gram TTR to LAS, the R^2 is at about 96%; in threshold selection for multi-source improvement, TTR scaling allows aggregate n-gram cosine similarity to reach an F1 score of about 94%.

Even considering that these numbers reflect performance on the same data used to create the models, these predictions are strong. While I expected sparsity measures to quantify "resourceiness" of corpora beyond basic length metrics, I didn't foresee TTR in particular being as effective as it turns out to be. It manages to capture corpus size effects as well as intrinsic complexity, as in the way Turkish is more complex than English in morphology and word order (see Figure 5.5). It's also a nicer metric than the number of word tokens in a corpus because the latter has a logarithmic relationship to LAS, as opposed to the linear relationship of TTR and LAS.

TTR is, of course, not a novel metric. It's usually only computed for word types and tokens, however; to my knowledge, this is the first work that has used TTR for distributions of n-grams. It's certainly the first work that has computed TTR for n-grams of oracle transitions.

If any one takeaway from this work is most immediately applicable, it's this. Distributions of n-gram frequencies for annotations are very informative about the composition of a corpus, and TTR is a particularly good metric for distilling this information. (While sequentially conditioned distributions might be stronger, it seems they aren't strictly necessary.) Its use does rest on the assumption than corpus splits (i.e., train and test) are both representative samples, and it does require some thought to mitigate brittleness when it comes to repetition (namely by only counting unique utterances/sentences), but for most natural language corpora, it's straightforward to apply.

8.2 Future Work vs. Applicability

Beyond n-gram TTR, the applicability of the current work comes with a number of caveats. Some of the issues are a matter of picking the right points to apply to other NLP projects; others are a matter of confounds and open questions that merit future work to address them.

Confounds and underlying factors

In terms of potential confounds, the linguistic nature of the annotations raises the question of the extent to which label-based metrics are useful because they're the same labels the models are learning vs. the extent to which they're useful because they capture the patterns of language data that models are likely to pick up on, explicitly labeled or not. To tease out these effects, I imagine a suite of experiments that vary by task, metrics, and the relationship between them. For example, using *named-entity recognition* (NER) and *language modeling* (LM) as possible different tasks:

- predict LAS and/or multi-source improvement, without POS tagging, using UD metrics
- predict NER or LM performance using UD metrics
- predict LAS and/or multi-source improvement using NER or LM metrics
- predict NER or sentiment analysis using metrics given those same labels

The first experiment is the most immediate extension: would the efficacy of *TTR-scaled POS n-gram cosine similarity* in predicting multi-source improvement be as strong without the parsers having learned POS tag labels explicitly? Or do POS tags represent a linguistic pattern in the data that models tend to learn regardless? The rest all pertain to how useful the UD annotation metrics are for language data even when modeling other tasks entirely – and vice versa.

Additional conditions, metrics, and methods

Other fairly immediate future work involves fleshing out the experimental space. The efficacy of the training set TTR baseline in predicting multi-source improvement suggests that I wasn't careful enough in balancing the multi-source conditions. More conditions with supplemental corpora that are dissimilar to the target corpora would help to balance things out.

Additionally, the surprising inefficacy of entropy-based metrics suggests that it would be worth looking into more traditional, sequentially-conditioned distributions. As effective as the feature-wise distributions are, I suspect that, especially for cross-corpus metrics, richer LM-style probabilities would be more so (as in, e.g., Rosa and Žabokrtskỳ, 2015; von Prince and Demberg, 2018).

Having determined that the metrics on their own are effective predictors using only regression and thresholds, it would be interesting to try more sophisticated methods of combining them into "real" models. Given the right method, the predictions could become extremely accurate – although this would certainly require some held-out data or cross-validation to avoid overfitting.

Unlabeled data and state-of-the-art models

Bringing the findings toward state-of-the-art NLP will require consideration of pretrained models. In the current experiments, I don't use unlabeled pretraining data – to keep the focus on *labeled* supplemental corpora, and in recognition of the fact that some languages are generally low-resource enough that even unlabeled data of the same language will be sparse.

Some amount of pretraining is standard practice in current NLP. Even the entries in Zeman et al.'s (2018) shared task were provided with additional unlabeled training data (which Che et al., 2018, for example, use for improving segmentation). Beyond pretraining with data just in the target language, deep contextual language models like multilingual BERT (Devlin et al., 2019) show that pretraining particularly large models on many monolingual corpora can serve as a strong foundation even for out-of-domain cases, like code-switching (Pires et al., 2019).

I think, however, that the ethos behind pretraining on unlabeled data of any sort is similar to the one behind the use of supplemental labeled corpora, and the same approach could be taken to account for both supplemental (labeled) corpora *and* (unlabeled) pretraining data. In the case that the pretraining data are accessible, they could be treated the same way as potential supplemental corpora, with metrics limited to those derived from the text (e.g., over word and character n-gram distributions). Alternatively, it could be interesting to explore the extent to which labels like those from UD (either gold-labeled or yielded for truly unlabeled data by a model) are effective for deriving metrics even when they are not actually being used alongside the text portion of the data.

In cases where the full data used to pretrain a model are either inaccessible or simply too large to deal with, calculating metrics would obviously be more difficult. It might still be possible to, say, derive predictions for how likely the target corpus is given the pretrained language model, and to use these probabilities as metrics.

In any case, pretrained models of any sort can be thought of in similar terms as supplemental corpora: they mitigate the sparsity of the target training data, depending on the similarity between the pretraining/supplemental data and the target corpus. Because of the overlap in sparsity-reducing effect, it's likely that access to good unlabeled data for pretraining would alleviate the need to rely on supplemental labeled data. Even so, it would be interesting to explore the interaction between the two, as well.

As for applicability of the work that's already done, I maintain that the exact numbers and thresholds are not the primary takeaway. Rather, the families of metrics – measurements over n-gram distributions, particularly TTR – and their connection to the models – using the same labels, but also going down to the character level, the same as the model input – are what's most important.

Thinking in terms of sparsity, complexity, and similarity; in terms of patterns both intrinsic to a corpus and relative to other corpora; in terms of using fairly simple corpus metrics to predict and analyze the behavior of what seem to be complex, black-box models, and to quantify what prior linguistic knowledge gets at intuitively – these are the main ingredients to the success of the predictive experiments, and, hopefully, to future experiments in a similar vein.

8.3 Returning the Favor to Linguistics

Throughout the experiments, I'm careful to maintain the perspective that all results pertain to *corpora*, not to *languages* directly. Inasmuch as they pertain to languages, it's that the corpora are representative samples of linguistic phenomena.

That being said, since the metrics are greatly inspired by literature in corpus linguistics, it's hard not to turn around and ask: *can these metrics tell us anything new about the languages in the corpora?* The answer is not straightforwardly *yes*, but that isn't to say there's no way for this sort of analysis to inform linguistics in the same way that linguistics informs the analysis.

For the most part, the metrics confirm and recreate matters of linguistic knowledge. Of course Turkish and German are useful in training a parser for Turkish-German code-switching, and of course Polish data are more helpful to Upper Sorbian than are Russian data. When the metrics yield misleading predictions, it's often because they've failed to capture patterns that seem obvious given prior linguistic knowledge. But the extent to which Maltese and Modern Standard Arabic have no recognizable similarities, or the relative similarities of the different branches of Germanic languages to English – these lead to accurate predictions that might have been missed using prior linguistic knowledge alone, and they could even point to avenues of worthwhile linguistic inquiry.

CHAPTER 9
Conclusion

In this dissertation, I set out to create and test a predictive methodology for concatenative training of dependency parsers. I use a mixture of established and novel metrics based on UD annotations, selecting the best among them for predicting both single-source performance and significant multi-source improvement.

The final predictive metrics are highly accurate, enabling not only the training of high-performing parsers, but also numerous avenues of analysis that are interesting both in terms of machine learning and in terms of linguistics. The metrics confirm my initial hypothesis: that multi-source improvement is proportional both to the similarity of the target and supplemental corpora and to the intrinsic sparsity and complexity of the target training corpus.

I'm hopeful about the applicability of the results – namely, in the usefulness of *label-wise n-gram type/token ratios* in characterizing the distributions essential to a corpus. There are a number of immediate extensions that could solidify the results further, including training more sophisticated models of prediction based on the metrics to build off of the simple regression- and threshold-based methods that already form a strong baseline. Early tests also suggest that the metrics could make predictions for zero-shot modeling, not just the concatenative models primarily explored in the current work.

The success in measuring and improving upon the low-resource languages of UD, in particular, has implications for what kinds of projects are most worth the cost of expert labeling. Even a few hundred tokens worth of linguistically labeled data are enough for metrics to pick out typological information, and for models to be able to bridge some of the gap between that sparse in-domain data and supplemental data from similar languages.

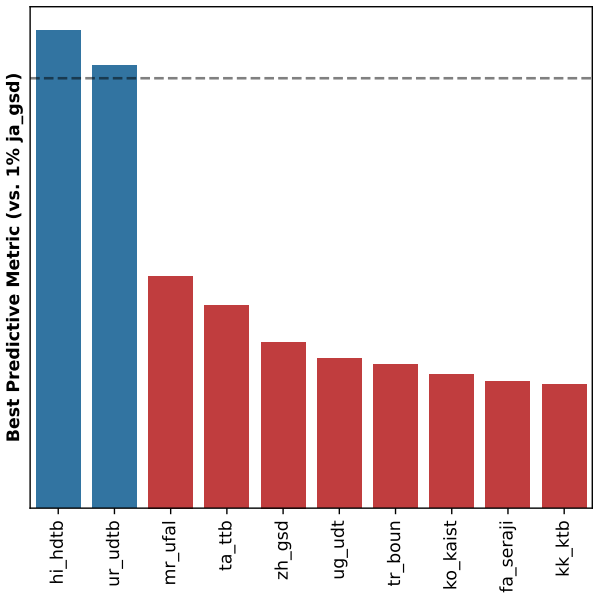


Figure 9.1: The top ten corpora in terms of predicted multi-source improvement for a 1% sample of a Japanese corpus. The dotted line represents the predictive threshold of significance. Blue bars indicate corpora that exceed the threshold, while red ones indicate corpora that fail to meet it.

Language-level typological labels have their uses, but even a small corpus of sentences with fine-grained, cross-linguistic labels can then be measured to empirically pick up on the same information, and then some. For the most part, these kinds of metrics quantify existing linguistic intuitions, rather than competing with them. Sometimes, however, they can point to cross-linguistic dynamics that linguistic inquiry is still exploring, – or has yet to explore – and that would be difficult for even an expert to predict.

Some of these dynamics are well-trodden, if not solved – such as the degree of similarity English has to the different branches of Germanic languages. Others are surprising, like the prediction in Figure 9.1 that the isolated Japanese language would, at a small enough corpus size, benefit most from additional data from Indic languages. Perhaps this is a pattern specific to the corpora, or perhaps it's a line of inquiry worth pursuing. Either way, the predictive methodologies I've put forth show that the cross-pollination of corpus linguistics and modern NLP can still be very fruitful.

BIBLIOGRAPHY

Aili, M., Mushajiang, W., Yibulayin, T., Abiderexiti, K., and Liu, Y. (2016). Universal dependencies for Uyghur. In *Proceedings of the Third International Workshop on Worldwide Language Service Infrastructure and Second Workshop on Open Infrastructures and Analysis Frameworks for Human Language Technologies (WLSI/OIAF4HLT2016)*, pages 44--50.

Ammar, W., Mulcaire, P., Ballesteros, M., Dyer, C., and Smith, N. A. (2016). Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431--444.

Ballesteros, M., Dyer, C., and Smith, N. A. (2015). Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015). 2015 September 17-21; Lisbon, Portugal.[Stroudsburg]: ACL, 2015. p. 349-59.* ACL (Association for Computational Linguistics).

Bentz, C., Ruzsics, T., Koplenig, A., and Samardzic, T. (2016). A Comparison Between Morphological Complexity Measures: Typological Data vs. Language Corpora. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 142--153.

Berdicevskis, A. and Bentz, C. (2018). Shared Task on Measuring Language Complexity.

Berg-Kirkpatrick, T., Burkett, D., and Klein, D. (2012). An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995--1005.

Bohnet, B. and Nivre, J. (2012). A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1455--1465. Association for Computational Linguistics.

Buchholz, S. and Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning (CoNLL-X)*, pages 149--164.

Candito, M. and Seddah, D. (2012). Le corpus Sequoia: annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *TALN 2012-19e conférence sur le Traitement Automatique des Langues Naturelles*.

Čéplö, S. (2018). Constituent order in Maltese: A quantitative analysis.

Çetinoğlu, Ö. and Çöltekin, Ç. (2019). Challenges of annotating a code-switching treebank. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 82--90.

Che, W., Liu, Y., Wang, Y., Zheng, B., and Liu, T. (2018). Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55--64.

Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740--750.

Collins, M. (1996). A New Statistical Parser Based on Bigram Lexical Dependencies. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 184--191.

Çöltekin, Ç. and Rama, T. (2018). Exploiting universal dependencies treebanks for measuring morphosyntactic complexity. *Proceedings of the First Shared Task on Measuring Language Complexity*, pages 1--8.

Cotterell, R., Mielke, S. J., Eisner, J., and Roark, B. (2018). Are All Languages Equally Hard to Language-Model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 536--541.

de Marneffe, M.-C., Manning, C. D., Nivre, J., and Zeman, D. (2021). Universal dependencies. *Computational Linguistics*, 47(2):255--308.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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 (Long and Short Papers)*, pages 4171--4186.

Dryer, M. S. and Haspelmath, M. (2013). *The world atlas of language structures online*. Max Planck Digital Library.

Duong, L., Cohn, T., Bird, S., and Cook, P. (2015). Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)*, pages 845--850.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334--343.

Eisner, J. M. (1996a). An Empirical Comparison of Probability Models for Dependency Grammar. Technical Report IRCS-96-11, Institute for Research in Cognitive Science, University of Pennsylvania.

Eisner, J. M. (1996b). Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 340--345. Association for Computational Linguistics.

Frinsel, F., Kingma, A., Swarte, F., and Gooskens, C. (2015). Predicting the asymmetric intelligibility between spoken Danish and Swedish using conditional entropy. *Tijdschrift voor Skandinavistiek*, 34(2).

Gal, Y. and Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019--1027.

Goldberg, Y. and Nivre, J. (2012). A dynamic oracle for arc-eager dependency parsing. In *Proceedings of COLING 2012*, pages 959--976.

Goldberg, Y. and Nivre, J. (2013). Training deterministic parsers with non-deterministic oracles. *Transactions of the association for Computational Linguistics*, 1:403--414.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602--610.

Gökrmak, M. and Tyers, F. M. (2017). A Dependency Treebank for Kurmanji Kurdish. In *Proceedings of the Fourth International Conference on Dependency Linguistics (DepLing, 2017)*, pages 64--73.

Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., and Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947.

Haspelmath, M., Dryer, M. S., Gil, D., and Comrie, B. (2005). *The world atlas of language structures*. Oxford Univ. Press.

Hatori, J., Matsuzaki, T., Miyao, Y., and Tsujii, J. (2011). Incremental joint POS tagging and dependency parsing in Chinese. In *Proceedings of 5th international joint conference on natural language processing*, pages 1216--1224.

Havelka, J. (2007). Beyond projectivity: Multilingual evaluation of constraints and measures on non-projective structures. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 608--615.

Hays, D. G. (1958). Grouping and dependency theories. In *Abstracts of the Conference on Machine Translation*.

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

Kingma, D. P. and Ba, J. L. (2015). Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Knuth, D. E. (1965). On the translation of languages from left to right. *Information and control*, 8(6):607--639.

Korenjak, A. J. (1969). A practical method for constructing LR (k) processors. *Communications of the ACM*, 12(11):613--623.

Kuhlmann, M., Gómez-Rodríguez, C., and Satta, G. (2011). Dynamic programming algorithms for transition-based dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 673--682.

Kuhlmann, M. and Nivre, J. (2006). Mildly non-projective dependency structures. In *Proceedings of the COLING/ACL 2006 main conference poster sessions*, pages 507--514.

Kulagina, O. S., Revzin, I. I., Moloshnaya, T., Volotskaya, Z. M., Paducheva, Y. V., Shelimova, I. N., and Shumilina, A. L. (1958). Various abstracts. In *Abstracts of the Conference on Machine Translation*. Translation by U. S. Joint Publications Research Service; original by First Moscow State Pedagogical Institute of Foreign Languages.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79--86.

Kyjánek, L. and Haviger, J. (2019). The Measurement of Mutual Intelligibility between West-Slavic Languages. *Journal of Quantitative Linguistics*, 26(3):205--230.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396--404.

Lin, D. (1995). A dependency-based method for evaluating broad-coverage parsers. *Proceedings of the International Joint Conference on Artificial Intelligence*.

Lin, Y.-H., Chen, C.-Y., Lee, J., Li, Z., Zhang, Y., Xia, M., Rijhwani, S., He, J., Zhang, Z., Ma, X., et al. (2019). Choosing Transfer Languages for Cross-Lingual Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125--3135.

Loshchilov, I. and Hutter, F. (2019). Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Marcus, M. P. (1978). *A theory of syntactic recognition for natural language.* PhD thesis, Massachusetts Institute of Technology.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523--530. Association for Computational Linguistics.

McDonald, R., Petrov, S., and Hall, K. (2011). Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 62--72.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153--157.

Mielke, S. J., Cotterell, R., Gorman, K., Roark, B., and Eisner, J. (2019). What Kind of Language Is Hard to Language-Model? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4975--4989.

Moberg, J., Gooskens, C., Nerbonne, J., and Vaillette, N. (2007). Conditional entropy measures intelligibility among related languages. In *Computational Linguistics in the Netherlands 2006: Selected papers from the 17th CLIN Meeting.*, pages 51--66. LOT.

Morerio, P., Cavazza, J., Volpi, R., Vidal, R., and Murino, V. (2017). Curriculum dropout. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3544--3552.

Naseem, T., Barzilay, R., and Globerson, A. (2012). Selective Sharing for Multilingual Dependency Parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 629--637.

Nivre, J. (2003). An Efficient Algorithm for Projective Dependency Parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149--160.

Nivre, J. (2004). Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50--57. Association for Computational Linguistics.

Nivre, J. (2009). Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 351--359. Association for Computational Linguistics.

Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659--1666.

Nivre, J., Kuhlmann, M., and Hall, J. (2009). An improved oracle for dependency parsing with online reordering. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 73--76.

Nivre, J. and Scholz, M. (2004). Deterministic dependency parsing of English text. In *Proceedings of the 20th international conference on Computational Linguistics*, page 64. Association for Computational Linguistics.

O'Neill, J. and Bollegala, D. (2018). Analysing Dropout and Compounding Errors in Neural Language Models. *arXiv preprint arXiv:1811.00998*.

Owens, J. (1988). *The foundations of grammar: an introduction to medieval Arabic grammatical theory*, volume 45. John Benjamins Publishing.

Park, H. H., Zhang, K. J., Haley, C., Steimel, K., Liu, H., and Schwartz, L. (2021). Morphology Matters: A Multilingual Language Modeling Analysis. *Transactions of the Association for Computational Linguistics*, 9:261--276.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024--8035. Curran Associates, Inc.

Pires, T., Schlinger, E., and Garrette, D. (2019). How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996--5001.

Rosa, R. and Žabokrtskỳ, Z. (2015). KLcpos3 – a Language Similarity Measure for Delexicalized Parser Transfer. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 243--249.

Ruder, S. and Plank, B. (2017). Learning to select data for transfer learning with Bayesian Optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372--382.

Santos, C. D. and Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1818--1826.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673--2681.

Seraji, M., Ginter, F., and Nivre, J. (2016). Universal Dependencies for Persian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2361--2365.

Smith, S., Kindermans, P.-J., Ying, C., and Le, Q. V. (2018). Don't decay the learning rate, increase the batch size. In *Proceedings of the 6th International Conference on Learning Representations*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929--1958.

Sulubacak, U., Gökırmak, M., Tyers, F., Çöltekin, Ç., Nivre, J., and Eryiğit, G. (2016). Universal dependencies for Turkish. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3444--3454.

Türk, U., Atmaca, F., Özateş, Ş. B., Berk, G., Bedir, S. T., Köksal, A., Başaran, B. Ö., Güngör, T., and Özgür, A. (2020). Resources for Turkish Dependency Parsing: Introducing the BOUN Treebank and the BoAT Annotation Tool. *arXiv preprint arXiv:2002.10416*.

von Prince, K. and Demberg, V. (2018). POS tag perplexity as a measure of syntactic complexity. *In Proceedings of the First Shared Task on Measuring Language Complexity*, pages 20--25.

Yamada, H. and Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 195--206.

Yang, L., Zhang, M., Liu, Y., Sun, M., Yu, N., and Fu, G. (2017). Joint POS tagging and dependence parsing with transition-based neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(8):1352--1358.

Yu, X. and Vu, N. T. (2017). Character Composition Model with Convolutional Neural Networks for Dependency Parsing on Morphologically Rich Languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 672--678.

Zeldes, A. (2017). The GUM Corpus: Creating Multilayer Resources in the Classroom. *Language Resources and Evaluation*, 51(3):581--612.

Zeman, D., Hajic, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J., and Petrov, S. (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.

Zeman, D., Nivre, J., Abrams, M., Ackermann, E., Aepli, N., Aghaei, H., Agić, Ž., Ahmadi, A., Ahrenberg, L., Ajede, C. K., Aleksandravičiūtė, G., Alfina, I., Antonsen, L., Aplonova, K., Aquino, A., Aragon, C., Aranzabe, M. J., Arıcan, B. N., Arnardóttir, Þ., Arutie, G., Arwidarasti, J. N., Asahara, M., Aslan, D. B., Ateyah, L., Atmaca, F., Attia, M., Atutxa, A., Augustinus, L., Badmaeva, E., Balasubramani, K., Ballesteros, M., Banerjee, E., Bank, S., Barbu Mititelu, V., Barkarson, S., Basmov, V., Batchelor, C., Bauer, J., Bedir, S. T., Bengoetxea, K., Berk, G., Berzak, Y., Bhat, I. A., Bhat, R. A., Biagetti, E., Bick, E., Bielinskienė, A., Bjarnadóttir, K., Blokland, R., Bobicev, V., Boizou, L., Borges Völker, E., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Boyd, A., Braggaar, A., Brokaitė, K., Burchardt, A., Candito, M., Caron, B., Caron, G., Cassidy, L., Cavalcanti, T., Cebiroğlu Eryiğit, G., Cecchini, F. M., Celano, G. G. A., Čéplö, S., Cesur, N., Cetin, S., Çetinoğlu, Ö., Chalub, F., Chauhan, S., Chi, E., Chika, T., Cho, Y., Choi, J., Chun, J., Cignarella, A. T., Cinková, S., Collomb, A., Çöltekin, Ç., Connor, M., Courtin, M., Cristescu, M., Daniel, P., Davidson, E., de Marneffe, M.-C., de Paiva, V., Derin, M. O., de Souza, E., Diaz de Ilarraza, A., Dickerson, C., Dinakaramani, A., Di Nuovo, E., Dione, B., Dirix, P., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eckhoff, H., Eiche, S., Eli, M., Elkahky, A., Ephrem, B., Erina, O., Erjavec, T., Etienne, A., Evelyn, W., Facundes, S., Farkas, R., Fernanda, M., Fernandez Alcalde, H., Foster, J., Freitas, C., Fujita, K., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Gerardi, F. F., Gerdes, K., Ginter, F., Godoy, G., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., González Saavedra, B., Griciūtė, B., Grioni, M., Grobol, L., Grūzītis, N., Guillaume, B., Guillot-Barbance, C., Güngör, T., Habash, N., Hafsteinsson, H., Hajič, J., Hajič jr., J., Hämäläinen, M., Hà Mỹ, L., Han, N.-R., Hanifmuti, M. Y., Hardwick, S., Harris, K., Haug, D., Heinecke, J., Hellwig, O., Hennig, F., Hladká, B., Hlaváčová, J., Hociung, F., Hohle, P., Huber, E., Hwang, J., Ikeda, T., Ingason, A. K., Ion, R., Irimia, E., Ishola, Ọ., Ito, K., Jelínek, T., Jha, A., Johannsen, A., Jónsdóttir, H., Jørgensen, F., Juutinen, M., K, S., Kaşıkara, H., Kaasen, A., Kabaeva, N., Kahane, S., Kanayama, H., Kanerva, J., Kara, N., Katz, B., Kayadelen, T., Kenney,

J., Kettnerová, V., Kirchner, J., Klementieva, E., Köhn, A., Köksal, A., Kopacewicz, K., Korkiakangas, T., Kotsyba, N., Kovalevskaitė, J., Krek, S., Krishnamurthy, P., Kuyrukçu, O., Kuzgun, A., Kwak, S., Laippala, V., Lam, L., Lambertino, L., Lando, T., Larasati, S. D., Lavrentiev, A., Lee, J., Lê Hồng, P., Lenci, A., Lertpradit, S., Leung, H., Levina, M., Li, C. Y., Li, J., Li, K., Li, Y., Lim, K., Lima Padovani, B., Lindén, K., Ljubešić, N., Loginova, O., Luthfi, A., Luukko, M., Lyashevskaya, O., Lynn, T., Macketanz, V., Makazhanov, A., Mandl, M., Manning, C., Manurung, R., Marşan, B., Mărănduc, C., Mareček, D., Marheinecke, K., Martínez Alonso, H., Martins, A., Mašek, J., Matsuda, H., Matsumoto, Y., Mazzei, A., McDonald, R., McGuinness, S., Mendonça, G., Miekka, N., Mischenkova, K., Misirpashayeva, M., Missilä, A., Mititelu, C., Mitrofan, M., Miyao, Y., Mojiri Foroushani, A., Molnár, J., Moloodi, A., Montemagni, S., More, A., Moreno Romero, L., Moretti, G., Mori, K. S., Mori, S., Morioka, T., Moro, S., Mortensen, B., Moskalevskyi, B., Muischnek, K., Munro, R., Murawaki, Y., Müürisep, K., Nainwani, P., Nakhlé, M., Navarro Horñiacek, J. I., Nedoluzhko, A., Nešpore-Bērzkalne, G., Nevaci, M., Nguyễn Thị, L., Nguyễn Thị Minh, H., Nikaido, Y., Nikolaev, V., Nitisaroj, R., Nourian, A., Nurmi, H., Ojala, S., Ojha, A. K., Olúòkun, A., Omura, M., Onwuegbuzia, E., Osenova, P., Östling, R., Øvrelid, L., Özateş, Ş. B., Özçelik, M., Özgür, A., Öztürk Başaran, B., Park, H. H., Partanen, N., Pascual, E., Passarotti, M., Patejuk, A., Paulino-Passos, G., Peljak-Łapińska, A., Peng, S., Perez, C.-A., Perkova, N., Perrier, G., Petrov, S., Petrova, D., Phelan, J., Piitulainen, J., Pirinen, T. A., Pitler, E., Plank, B., Poibeau, T., Ponomareva, L., Popel, M., Pretkalniņa, L., Prévost, S., Prokopidis, P., Przepiórkowski, A., Puolakainen, T., Pyysalo, S., Qi, P., Rääbis, A., Rademaker, A., Rama, T., Ramasamy, L., Ramisch, C., Rashel, F., Rasooli, M. S., Ravishankar, V., Real, L., Rebeja, P., Reddy, S., Rehm, G., Riabov, I., Rießler, M., Rimkutė, E., Rinaldi, L., Rituma, L., Rocha, L., Rögnvaldsson, E., Romanenko, M., Rosa, R., Roca, V., Rovati, D., Rudina, O., Rueter, J., Rúnarsson, K., Sadde, S., Safari, P., Sagot, B., Sahala, A., Saleh, S., Salomoni, A., Samardžić, T., Samson, S., Sanguinetti, M., Sanıyar, E., Särg, D., Saulīte, B., Sawanakunanon, Y., Saxena, S., Scannell, K., Scarlata, S., Schneider, N., Schuster, S., Schwartz, L., Seddah, D., Seeker, W., Seraji, M., Shen, M., Shimada, A., Shirasu, H., Shishkina, Y., Shohibussirri, M., Sichinava, D., Siewert, J., Sigurðsson, E. F., Silveira, A., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Skachedubova, M., Smith, A., Soares-Bastos, I., Spadine, C., Sprugnoli, R., Steingrímsson, S., Stella, A., Straka, M., Strickland, E., Strnadová, J., Suhr, A., Sulestio, Y. L., Sulubacak, U., Suzuki, S., Szántó, Z., Taji, D., Takahashi, Y., Tamburini, F., Tan, M. A. C., Tanaka, T., Tella, S., Tellier, I., Testori, M., Thomas, G., Torga, L., Toska, M., Trosterud, T., Trukhina, A., Tsarfaty, R., Türk, U., Tyers, F., Uematsu, S., Untilov, R., Urešová, Z., Uria, L., Uszkoreit, H., Utka, A., Vajjala, S., van der Goot, R., Vanhove, M., van Niekerk, D., van Noord, G., Varga, V., Villemonte de la Clergerie, E., Vincze, V., Vlasova, N., Wakasa, A., Wallenberg, J. C., Wallin, L., Walsh, A., Wang, J. X., Washington, J. N., Wendt, M., Widmer, P., Williams, S., Wirén, M., Wittern, C., Woldemariam, T., Wong, T.-s., Wróblewska,

A., Yako, M., Yamashita, K., Yamazaki, N., Yan, C., Yasuoka, K., Yavrumyan, M. M., Yenice, A. B., Yıldız, O. T., Yu, Z., Žabokrtský, Z., Zahra, S., Zeldes, A., Zhu, H., Zhuravleva, A., and Ziane, R. (2021). Universal Dependencies 2.8.1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Zeman, D., Popel, M., Straka, M., Hajic, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., et al. (2017). CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1--19.

Zeman, D. and Resnik, P. (2008). Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.

Zhai, Z., Nguyen, D. Q., and Verspoor, K. (2018). Comparing CNN and LSTM character-level embeddings in BiLSTM-CRF models for chemical and disease named entity recognition. *EMNLP 2018*, page 38.

Zhang, Y. and Weiss, D. (2016). Stack-propagation: Improved Representation Learning for Syntax. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1557--1566.

Zipf, G. (1935). *The psycho-biology of language: an introduction to dynamic philology*. Houghton Mifflin.

APPENDIX A

Full Results & Data

A.1 Single-Source Models

TARGET CORPUS	UAS	LAS	POS
ar_padt	85.36	80.27	96.24
bxr_bdt	32.58	15.28	47.96
de_gsd	84.69	79.60	93.96
en_gum	85.78	82.42	95.78
en_lines	84.53	80.14	95.25
en_partut	87.06	83.45	94.69
fa_seraji	87.06	82.95	96.44
fi_ftb	86.55	82.47	94.23
fr_partut	88.63	85.25	95.70
fr_sequoia	88.91	86.51	97.63
hsb_ufal	39.46	22.54	48.76
it_isdt	91.53	89.29	97.73
it_partut	88.49	85.59	96.93
ja_gsd	93.43	91.93	97.43
kk_ktb	48.30	27.45	54.71
kmr_mg	39.62	23.08	53.69
ko_gsd	84.13	80.15	94.85
ko_kaist	86.42	84.19	94.64

TARGET CORPUS	UAS	LAS	POS
lt_alksnis	72.99	66.74	91.17
lt_hse	50.09	34.91	73.21
mr_ufal	72.82	56.55	80.34
mt_mudt	80.60	74.16	93.56
nl_lassysmall	86.97	82.17	94.36
olo_kkpp	30.17	13.24	44.49
pl_pdb	90.43	87.34	97.99
qtd_sagt	63.36	50.38	85.25
ru_gsd	86.59	81.75	96.52
sv_lines	84.98	80.30	94.89
sv_talbanken	86.93	82.20	96.45
ta_ttb	68.53	55.81	81.00
te_mtg	88.90	78.78	90.57
tr_boun	75.51	68.23	91.59
tr_imst	69.58	62.49	93.67
ug_udt	75.67	63.52	87.45
zh_gsd	77.40	73.04	92.57

Table A.1: Results of single-source training for each target corpus.

TARGET CORPUS	SIZE	UAS	LAS	POS
en_gum	5K (5%)	68.64	60.06	83.13
en_gum	10K (10%)	73.28	66.57	88.19
en_gum	51K (50%)	82.15	78.04	94.36
mt_mudt	2K (10%)	64.41	53.37	82.22
ug_udt	981 (5%)	61.73	41.45	73.48
ug_udt	2K (10%)	63.66	46.64	77.81
ug_udt	9K (50%)	72.88	59.25	85.93

Table A.2: Results of subsampled single-source training for each target corpus.

A.2 Multi-Source Models

Target Corpus	Supplemental Corpus	UAS	LAS	POS
bxr_bdt	kk_ktb	42.68	18.60	49.74
	ug_udt	34.12	16.01	44.83
de_gsd	qtd_sagt	84.63	79.57	94.01
	qtd_sagt, tr_boun	83.48	78.31	93.61
	qtd_sagt, tr_imst	84.44	79.41	93.80
en_gum	en_lines	86.00	82.40	95.15
	qtd_sagt	85.44	81.91	95.48
en_lines	en_gum	86.12	82.53	94.70
	nl_lassysmall	84.24	79.71	95.16
	en_partut	85.38	81.10	95.19
	sv_lines	84.55	79.93	94.81
en_partut	en_lines	87.65	84.24	94.78
	fr_partut	86.74	83.30	94.75
	mt_mudt	86.09	82.95	94.54
fa_seraji	kmr_mg	86.96	82.67	96.41
	ug_udt	86.37	82.25	96.42
fr_partut	en_partut	88.59	85.13	95.31
	fr_sequoia	89.44	86.29	96.35
fr_sequoia	fr_partut	90.16	87.50	97.67
hsb_ufal	cs_cltt	41.88	33.19	67.99
	cs_fictree	44.31	36.56	72.64
	ru_gsd	38.28	28.96	59.72
it_partut	mt_mudt	88.98	85.91	96.92
kk_ktb	bxr_bdt	48.85	26.76	53.72
	tr_boun	47.37	28.75	58.48
	ug_udt	53.04	33.30	61.08
kmr_mg	fa_seraji	40.15	26.33	58.46
mt_mudt	en_partut	80.45	74.71	93.99
	it_partut	81.15	74.93	93.89
nl_lassysmall	en_lines	86.75	82.03	94.36
qtd_sagt	en_gum	62.30	49.21	84.87
	de_gsd	65.21	52.41	86.44
	de_gsd, tr_boun	67.44	55.95	88.31
	de_gsd, tr_imst	68.42	56.66	88.68
	tr_boun	64.44	53.02	86.91
	tr_imst	64.31	52.26	86.31
ru_gsd	hsb_ufal	86.15	81.34	96.48
sv_lines	en_lines	85.35	80.48	94.92
	sv_talbanken	86.84	82.30	95.63
sv_talbanken	sv_lines	86.05	82.17	96.73
ta_ttb	te_mtg	69.33	58.12	82.70
te_mtg	ta_ttb	89.88	79.61	91.12
tr_boun	de_gsd, qtd_sagt	74.74	67.06	90.85
	kk_ktb	74.85	67.41	91.65
	tr_imst	75.70	67.79	91.67
	qtd_sagt	75.16	67.49	91.33
	ug_udt	75.81	68.25	91.56
tr_imst	de_gsd, qtd_sagt	68.88	60.29	91.97
	tr_boun	71.58	61.33	93.63
	qtd_sagt	69.46	61.64	92.99
	ug_udt	69.31	62.02	92.64
ug_udt	bxr_bdt	75.03	62.56	87.82
	fa_seraji	75.15	62.26	87.65
	kk_ktb	74.82	62.47	87.28
	tr_boun	76.53	63.63	87.92
	tr_imst	76.07	63.56	88.17

Table A.3: Results of multi-source training (without fine-tuning) for each target corpus.

Target Corpus	Supplemental Corpus	UAS	LAS	POS
bxr_bdt	kk_ktb	40.74	18.78	49.71
	ru_gsd	25.01	9.70	46.09
	ug_udt	40.88	18.49	50.88
en_gum	en_lines	86.05	82.77	95.70
	sv_lines	85.68	82.01	95.61
en_lines	en_gum	86.92	83.34	96.07
	nl_lassysmall	85.01	80.76	95.40
	sv_lines	85.45	80.97	95.09
en_partut	fr_partut	86.53	82.69	94.22
fr_partut	en_partut	89.17	85.98	95.74
hsb_ufal	cs_cltt	43.46	34.37	68.52
	cs_fictree	46.12	38.20	74.07
	en_gum	39.33	24.61	60.14
	pl_lfg	49.12	41.29	79.64
	pl_pdb	55.87	46.54	78.14
	ru_gsd	40.98	30.72	60.99
kk_ktb	bxr_bdt	51.64	28.99	54.64
	tr_boun	51.15	32.41	59.51
	ru_gsd	36.04	18.70	54.78
	ug_udt	50.01	30.90	57.31
kmr_mg	fa_seraji	42.65	29.78	59.48
	mt_mudt	32.07	19.18	51.70
	tr_boun	32.93	17.31	50.77
lt_hse	lt_alksnis	63.40	51.89	85.47
	lv_lvtb	53.77	39.06	78.49
	mt_mudt	45.85	31.89	75.00

Target Corpus	Supplemental Corpus	UAS	LAS	POS
mt_mudt	ar_padt	80.57	73.99	93.71
	en_partut	80.38	74.18	93.76
	it_isdt	81.06	74.97	93.39
	it_partut	80.76	74.69	93.82
	fa_seraji	80.10	73.67	93.99
olo_kkpp	fi_ftb	48.19	30.44	69.62
	fr_partut	31.79	15.12	45.63
qtd_sagt	en_gum	62.97	49.98	85.31
	de_gsd	67.25	54.26	87.14
	de_gsd, tr_boun	69.56	58.86	90.31
	de_gsd, tr_imst	69.75	58.34	89.77
	tr_boun	64.80	53.53	87.39
	tr_imst	65.69	53.63	87.45
sv_lines	en_lines	85.67	80.65	94.83
	sv_talbanken	86.84	82.63	95.97
sv_talbanken	sv_lines	87.25	83.93	97.16
ta_ttb	te_mtg	68.38	56.11	81.90
te_mtg	ta_ttb	88.63	78.92	91.40
tr_boun	tr_imst	75.48	68.04	91.78
tr_imst	tr_boun	72.83	66.35	94.64
ug_udt	fa_seraji	75.26	62.18	87.33
	ko_kaist	74.83	62.27	86.49
	tr_boun	76.57	63.59	87.47
	tr_imst	75.09	62.64	87.35

Table A.4: Results of multi-source training (with fine-tuning) for each target corpus.

Target Corpus	Size	Supplemental Corpus	UAS	LAS	POS
en_gum	5K (5%)	en_lines	79.17	73.83	91.12
		sv_lines	70.90	63.39	86.36
en_gum	10K (10%)	nl_lassysmall	74.54	67.93	89.38
		en_lines	84.65	80.36	94.01
		fr_sequoia	75.65	69.28	89.73
		sv_lines	76.08	69.47	89.39
		tr_boun	74.11	66.22	90.83
en_gum	51K (50%)	en_lines	84.37	80.52	94.89
		sv_lines	83.40	79.34	94.03
mt_mudt	2K (10%)	it_isdt	66.34	55.01	81.33
ug_udt	981 (5%)	tr_boun	60.73	44.27	75.55
ug_udt	2K (10%)	fa_seraji	63.35	44.23	77.02
		tr_boun	65.68	49.56	78.15
ug_udt	9K (50%)	tr_boun	74.67	60.76	86.09

Table A.5: Results of subsampled multi-source training (with fine-tuning) for each target corpus.

A.3 Corpus Domains

Corpus	Genres
ar_padt	news
bxr_bdt	fiction, grammar-examples, news
cs_cltt	legal
cs_fictree	fiction
de_gsd	news, reviews, wiki
en_gum	academic, blog, fiction, government, news, nonfiction, social, spoken, web, wiki
en_lines	fiction, nonfiction, spoken
en_partut	legal, news, wiki
fa_seraji	fiction, legal, medical, news, nonfiction, social, spoken
fi_ftb	grammar-examples
fi_tdt	blog, fiction, grammar-examples, legal, news, wiki
fr_partut	legal, news, wiki
fr_sequoia	medical, news, nonfiction, wiki
hsb_ufal	nonfiction, wiki
it_isdt	legal, news, wiki
it_partut	legal, news, wiki
ja_gsd	blog, news
kk_ktb	fiction, news, wiki
kmr_mg	fiction, wiki
ko_gsd	blog, news
ko_kaist	academic, fiction, news
lt_alksnis	fiction, legal, news, nonfiction
lt_hse	news, nonfiction
lv_lvtb	academic, fiction, legal, news, spoken
mr_ufal	fiction, wiki
mt_mudt	fiction, legal, news, nonfiction, wiki
nl_lassysmall	wiki
olo_kkpp	news, nonfiction, web
pl_lfg	fiction, news, nonfiction, social, spoken
pl_pdb	fiction, news, nonfiction
qtd_sagt	spoken
ru_gsd	wiki
sv_lines	fiction, nonfiction, spoken
sv_talbanken	news, nonfiction
ta_ttb	news
te_mtg	grammar-examples
tr_boun	news, nonfiction
tr_imst	news, nonfiction
ug_udt	fiction
zh_gsd	wiki

Table A.6: Source genres for the full set of corpora used for calculating metrics and training parsers in the current work. All corpora come from the UD 2.8 collection (Zeman et al., 2021). Basic information for each corpus is listed in the corresponding Table 3.3.