On the Relation between Syntactic Divergence and Zero-Shot Performance

Ofir Arviv[†*] **Dmitry Nikolaev**[‡*] **Taelin Karidi**[†] **Omri Abend**[†]
[†]Hebrew University of Jerusalem [‡]Institute for Natural Language Processing, University of Stuttgart
{ofir.arviv|taelin.karidi|omri.abend}@mail.huji.ac.il
dnikolaev@fastmail.com

Abstract

We explore the link between the extent to which syntactic relations are preserved in translation and the ease of correctly constructing a parse tree in a zero-shot setting. While previous work suggests such a relation, it tends to focus on the macro level and not on the level of individual edges—a gap we aim to address. As a test case, we take the transfer of Universal Dependencies (UD) parsing from English to a diverse set of languages and conduct two sets of experiments. In one, we analyze zero-shot performance based on the extent to which English source edges are preserved in translation. In another, we apply three linguistically motivated transformations to UD, creating more cross-lingually stable versions of it, and assess their zero-shot parsability. In order to compare parsing performance across different schemes, we perform extrinsic evaluation on the downstream task of cross-lingual relation extraction (RE) using a subset of a popular English RE benchmark translated to Russian and Korean.[1] In both sets of experiments, our results suggest a strong relation between cross-lingual stability and zero-shot parsing performance.

1 Introduction

Recent progress in cross-lingual transfer methods, such as multi-lingual embeddings (Devlin et al., 2018; Mulcaire et al., 2019), enabled significant advances in a wide range of cross-lingual natural language processing tasks. The transferred models, however, are not uniformly effective in addressing languages with different grammatical structures, and little is known about the settings under which cross-lingual transfer is more or less effective.

A prominent way of facilitating transfer of grammatical knowledge from one language to another is through the use of cross-lingual symbolic representation schemes (Chen et al., 2017, 2018; Bugliarello and Okazaki, 2020). Many advances have been made in this area in recent years, most notably the development and quick adoption of Universal Dependencies (UD; Nivre et al., 2016), a cross-lingually applicable scheme that has become the de facto standard for syntactic annotation.

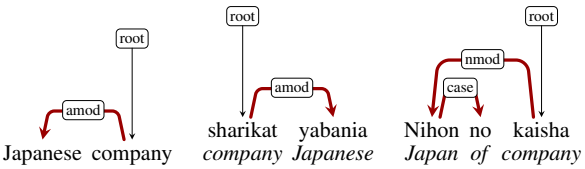

Figure 1: A UD parse of the English phrase *Japanese company* (left) and of its translations to Arabic (middle) and Japanese (right). Word order aside, the UD tree of the Arabic translation is identical to the English one while that of the Japanese translation is different. The `amod` edge in this phrase is stable in translation to Arabic but unstable in translation to Japanese.

While these schemes abstract away from many syntactic differences, there is still considerable variability in the strategies employed to express the same basic meanings across languages. In this work, we are mainly interested in the flip side of variability, viz. the *stability* of a given scheme—the extent to which its annotations are invariant under translation. See an example in Fig. 1.

We present two sets of empirical findings that establish a strong relation between stability and success of zero-shot (ZS) cross-lingual transfer from a single language in a multiply parallel corpus setting where no annotated data from the target languages were used for training. Such a setting is a natural starting point for our investigation, as it is both practically useful (see, e.g., Ammar et al., 2016; Schuster et al., 2019; Wang et al., 2019; Xu and Koehn, 2021, for successful examples of employing ZS learning cross-lingually) and is based on

[*]Equal contribution. Dmitry Nikolaev's work was undertaken during his post-doc at Stockholm University.

[1]All resources are available at https://github.com/OfirArviv/translated_tacred and https://github.com/OfirArviv/improving-ud

a homogeneous training set, which minimizes the risk of introducing confounds into the analysis.

The first set of experiments quantifies the effect of stability of edges in UD parse trees on a ZS parser's performance. In order to check if a particular edge was stably transferred from the corresponding source sentence, we use an extended version of the manually aligned subset of the Parallel UD dataset (Zeman et al., 2017a; Nikolaev et al., 2020), which provides word-aligned translations of circa 1000 English sentences into six languages with different typological profiles. We find highly consistent trends across all language pairs, where stable edges receive an average labeled attachment score (LAS; a standard evaluation metric in dependency parsing) that is often twice or more bigger than the one for edges that do not correspond to a source side edge. These findings indicate a strong link between the stability of an edge and its contribution to ZS parsing quality and suggest that ZS parsing performance can be enhanced by improving the stability of the underlying syntactic representation.

Our next set of experiments are the first steps in this direction. Concretely, we define three transformations, each targeting an area of syntax that is known to give rise to cross-lingual divergences in UD terms, and thus create three slightly modified versions of UD. We then apply these transformations to the training set in order to check if these versions of UD lead to improved ZS performance. As attachment scores across different schemes are not comparable, we opt for extrinsic evaluation through ZS cross-lingual relation extraction.

Since there are no multilingual RE datasets that include data from non-Western-European languages, we translated a 500-sentence subset of the English TACRED dataset (Zhang et al., 2017) into Korean and Russian and annotated it with the relations from respective English source sentences. Our results show that modified versions of UD give rise to improved results. This indicates that UD can be made more cross-lingually stable without sacrificing its usefulness for downstream tasks.

The paper is organized as follows. In §2, we explore the correlation between ZS performance and stability. In §3, we present transformations to UD annotations (§3.1) and the methodology for comparing downstream usefulness of vanilla UD and transformed UD (§3.2). Our experimental setup is described in §4, and the results are presented in §5. Related work is summarised in §6. Section 7 concludes the paper.

2 Edge Stability and ZS Parsability

This section evaluates the relation between the extent to which an edge in a translated sentence corresponds to an edge in the original sentence (which we operationalize as several *stability categories*) and the ability of a ZS parser to parse it correctly (its *ZS parsability*). We use the manually aligned subset of the Parallel UD corpus (N20; Nikolaev et al., 2020), which augments the PUD dataset (Zeman et al., 2017b) with alignments between corresponding content words over five language pairs, the source language being English (En), and the target languages being French (Fr), Russian (Ru), Japanese (Jp), Chinese (Zh), and Korean (Ko). We further use an extension of the corpus to an additional language pair, English-Arabic (En-Ar), in order to increase the diversity of examined languages (Rafaeli et al., 2021).[2] In all cases, both the UD annotation and the alignment were done manually.

We next train a state-of-the-art ZS parser on En (same as used for the experiment described in §3.2) and examine its performance on PUD over the six target languages. We partition the edges in the target-language test parses into categories based on their stability and investigate the performance of the parser on an edge as a function of its stability-category membership.

2.1 Experimental Setup

Let S_e be a UD annotated sentence in the source language and S_l its translation in the target language. Let (w'_1, w'_2) be a pair of words in S_l and (w_1, w_2) the corresponding aligned words in S_e, if such exist. We partition the edges in S_l according to the following scheme:

1. *Fully Aligned* edges are e'=(w'_1, w'_2) in S_l between two content words with label l, such that there is an edge $e = (w_1, w_2)$ is in S_e with label l.

2. *Partially Aligned* edges are $e' = (w'_1, w'_2)$ in S_l between two content words with label l, such that there is $e = (w_1, w_2)$ is in S_e with label $l' \neq l$.

[2]The annotation was carried out by a single annotator, proficient in English and Arabic, using the same guidelines as for the original aligned PUD corpora.

3. *Unaligned* edges are $e = (w'_1, w'_2)$ in S_l between two content words where either w'_1 or w'_2 do not have a single aligned word in S_e.

4. *Flipped* edges are $e' = (w'_1, w'_2)$ in S_l between two content words, such that in S_e there exists an edge $e = (w_2, w_1)$.

5. *Misaligned* edges are $e' = (w'_1, w'_2)$ in S_l between two content words, where w'_1 and w'_2 have aligned words in S_e that are not an fully/partially aligned or flipped edges.

6. *Function Word* edges are $e' = (w'_1, w'_2)$ in S_l where either w'_1 or w'_2 is a function word.[3]

We note that each edge belongs to exactly one of the above categories. We report LAS and unlabeled attachment scores (UAS) for each edge category, averaged over 10 runs. Fully Aligned edges is the most stable category, whereas Misaligned and Flipped edges constitute the least stable one because relying on the edge or path connecting the corresponding words in the original tree would be harmful for performance. Partially Aligned edges represent a special case: this is the second-most stable category in terms of tree structure (UAS) but the least stable one in terms of labels (LAS).

2.2 Results and Discussion

Our main results are presented in Table 1. Standard deviations and the percentages of edge categories can be found in Appendix A.1. We find that the ZS parsability of an edge strongly correlates with its stability. In fact, for UAS, we find that the ZS parsability is almost invariably ordered in the following way:

Fully Aligned > Partially Aligned > Unaligned > Misaligned > Flipped

For LAS, the ordering is similar, except that Partially Aligned is occasionally positioned lower and Misaligned and Flipped swap places in En-Jp.

Moreover, the differences between the scores for the different categories are substantial, with the most stable categories generally obtaining about two times the labeled score of the least stable ones and seeing a 60% increase in unlabeled scores. We can also see a substantial difference between the scores of the Fully Aligned and Unaligned Token edges: around 10–20% and 20–40% increase in unlabeled and labeled performance, respectively, for structurally similar languages such as En-Ru, and around $30\%+$ (UAS) and $55\%+$ (LAS) increase for more divergent language pairs. This suggests that, despite recent advances in ZS parsing technology, performance is still dependent on the extent to which parallel constructions are prevalent in the source language.

These results lend strong support to our hypothesis that the stability of an edge affects its ZS parsability. They also suggest that when evaluating ZS models, it is informative to distinguish between edges belonging to the most and least stable categories as these categories may pose different challenges that may benefit from the application of different methods. For example, some methods may improve the implicit alignment capabilities of the employed multilingual embeddings while others may improve the parser's ability to abstract away from surface differences and correctly predict unaligned edges.

Function Word edges contain edges of varying stability: some are aligned, while others are not. We thus expect the parser performance on this type of edge to be better than on Flipped and Misaligned edges but lower than on Fully Aligned. Indeed, the results match our expectations, with the exception of LAS in En-Ar, where the LAS on the Function Word edges is higher than that on Fully Aligned edges. In En-Ko and En-Ja, Flipped edges and Unaligned edges, respectively, achieve higher UAS then Function Word edges, but the differences are within one standard deviation.

Standard deviations (over 10 runs) on all edge and score types are small, usually less than 2. The prevalence of each edge type varies depending on the target language's similarity to English, but Partially Aligned and Misaligned edges usually constitute about 4–8% each and Flipped Edges about 1.3–3%, indicating that these difficult cases are present in all languages. For the full data see Appendix A.1.

We note that Partially Aligned edges present considerably lower LAS than Fully Aligned ones. Inspecting the Partially Aligned edges that were incorrectly predicted, we see that the parser has a strong bias towards predicting source-side labels. This tendency accounts for 46% and 42% of the errors, respectively, in En-Ru and En-Fr, and for 19%,

[3] Function words were not aligned to other function words in the PUD corpus as such an alignment was deemed unreliable (see N20). Some Function Word edges could be aligned to edges in S_e, but this number is low for highly morphosyntactically divergent language pairs.

Edge Type	Russian		French		Chinese		Japanese		Korean		Arabic	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Fully Aligned	88	83	88	82	61	48	52	37	54	37	67	50
Partially Aligned	82	41	83	56	56	30	53	24	53	22	63	22
Unaligned	74	59	78	67	44	28	30	10	41	23	52	32
Misaligned	54	45	55	49	28	16	22	11	34	13	39	25
Flipped	48	29	54	45	26	13	19	10	37	12	41	23
Func Word	68	62	79	72	36	25	20	13	36	20	65	58

Table 1: UD zero-shot performance on the PUD corpora per edge category (averaged over 10 models). Rows correspond to stability categories; columns, to target languages and score types. See §2 for the definitions of stability categories.

21%, 20% and 31% of the errors, respectively, in En-Zh, En-Ja, En-Ko, and En-Ar. These results suggest that defining relations that are less likely to be altered in translation can substantially improve a scheme's transferability, a direction we explore in the following sections.

Last, in order to gain a better insight into the edges constituting each of the stability categories, we analyze the performance of a supervised parser on them (see Appendix A.2 for the training setup). We find that, surprisingly, the supervised parsability of an edge strongly correlates with its cross-lingual stability: the parser's performance on different stability categories is generally ordered in the same way as in the ZS setting. This finding suggests that cross-lingual stability has a connection to the ability of the parser to generalize within a language as well, a direction which we defer to future work.

It must be noted, however, that the performance difference between the categories is less pronounced in the supervised setting than in the ZS setting, which lends support to our hypothesis as to the relation between stability and ZS parsability. For more details, see Appendix A.2.

To summarize, our analysis shows that ZS parsing performance is better for edges that closely correspond to similar constructions in English. However, while performance increases with stability even for highly similar language pairs (En-Fr and En-Ru), the scores still lag behind their supervised counterparts, which may suggest that the underlying cross-lingual embeddings can be improved.

3 Comparing Zero-shot Performance across Representation Schemes

In this section, we aim to manipulate the annotation scheme so as to increase its stability and test whether the modification yields more cross-lingually useful annotations. We achieve this by devising three linguistically motivated, language agnostic transformations (§3.1) and applying them to the downstream task of ZS relation extraction (§3.2). While some previous work proposed syntactic preprocessing of the source-side UD trees for the sake of cross-lingual ZS parsing, we are not aware of any previous work that compares the performance on a parsing-dependent downstream task across different schemes.

3.1 Transformations

We devise three linguistically motivated, universally applicable transformations, based on linguistic typological literature and the findings of N20. The aim of the transformations is to abstract away from syntactic distinctions made by UD that are unstable and of low information value and thus to improve cross-lingual transfer. These transformations bear a conceptual resemblance to the transformations explored in (Ponti et al., 2018). Those, however, are tailored to specific language pairs, while ours are applicable to any language pair. See §6 for further discussion.

For simplicity, we explore transformations that only alter edge labels and preserve the tree topology and defer transformations that alter the tree topology to future work. In terms of the analysis presented in §2, our transformations aim to increase the congruence between source and target sentences by converting Partially Aligned edges to Fully Aligned ones.

3.1.1 Normalization of Nominal Modification

Examining the alignment matrices from N20, we can see that in many languages, `amod`, `acl`, `nmod`, and `compound` form more or less a complete graph of what they may be aligned with in different languages, which corresponds to observations made by linguists that languages have different patterns of nominal modification (Maniez, 2012; García, 2006).

We hypothesize therefore that there could be a

benefit in representing nominal modification using a simplified, more general scheme. The transformation NOMINAL therefore converts UD `compound`, `nmod`, and `amod` into `acl`.

3.1.2 Normalization of Nominal Predicates

One of the sources of cross-lingual discrepancies in UD annotations of translated sentences, as detected by N20's data, is the handling of nominal predicates, which UD does not distinguish from other nouns. For example, in *the king's hawk* and *the king's death*, *king* will be labeled as a UD `nmod` in both cases, although in the first case it is semantically a possessor, while in the second case it is a subject of a change of state. When translated or rephrased, *the king's death* may end up either as a structure that parallels the source or as a verb-headed clause, similar to *the king died*. See Figure 2 for a cross-lingual example.

Therefore, UD's lumping of predicative and non-predicative nouns is a potential source of divergence. In order to arrive at a more stable handling of predicate nominals, we employ UCCA (Abend and Rappoport, 2013), a semantic representation that explicitly distinguishes between the two structures. We use TUPA (Hershcovich et al., 2018) to parse the training corpus, identify subtrees headed by Processes, and relabel them as subordinate clauses. For more details see Appendix D.1. We use PREDICATE to refer to this transformation.

3.1.3 Normalization of Obliques

One of the least clear elements of the UD annotation guidelines is the distinction between `obl` (obliques), defined as "non-core (oblique) arguments or adjuncts", and `iobj`, defined as "any nominal phrase that is a core argument of the verb but is not its subject or (direct) object". The latter definition is clarified as referring mostly to arguments of "ditransitive verbs of exchange", and the clarification of the former essentially equates is with adjuncts.[4] This leaves a lot of ambiguous cases (is *from his friend* in *He got a telephone call from his friend* an argument of a ditransitive verb of exchange?), and in practice the distinction boils down to whether there is a case-marking token: PPs are routinely treated as `obl`, and NPs with non-nominative/accusative morphological case marking are treated as `iobj`.

[4] "This means that it functionally corresponds to an adverbial attaching to a verb, adjective or other adverb."

This leads to a lot of spurious discrepancies between languages: English corpora usually have only a handful of `iobj`, while languages with rich case systems, such as Russian, have them in abundance. Moreover, `obl` in some languages may often correspond to `advmod` in others, which is unsurprising given their overt semantic connection.

In order to bridge these divergences, we propose to retire the `obl` category altogether, because in most cases it does not provide any useful information in addition to the fact that there is a nominal-headed subtree with a case-marking token. We propose to split all `obl` into `advmod` and `iobj` based on their semantics, which we recover by applying the SNACS preposition supersense parser (Liu et al., 2020b)[5] to the input. `obl` with SNACS tags largely corresponding to typical adverbial semantics[6] are converted to `advmod`, and all others are converted to `iobj`. We use OBLIQUE to refer to this transformation.

3.2 Application to ZS relation extraction

Comparing cross-lingual transferability of different annotation schemes presents a methodological problem: simply comparing the LAS/UAS obtained by a ZS parser may be misleading as the increased performance may come at the expense of losing useful semantic distinctions.

We therefore opt for extrinsic evaluation using the task of ZS cross-lingual relation-extraction on the TACRED dataset (Zhang et al., 2017). Concretely, we take a pattern-based approach to RE, applying the setup that Tiktinsky et al. (2020) used for monolingual RE to ZS cross-lingual transfer. RE is a natural choice for this experiment as it is both a core task in NLP, and one that is dependent on syntactic annotations (at least in some state-of-the-art approaches).

Despite the importance of RE, to our knowledge, all existing datasets for this and similar tasks only target Western European languages, which generally resemble English in their grammatical structure. We therefore translate and annotate a little more than 500 examples from the TACRED dataset into Russian and Korean and use these examples as test sets (see Appendix B for details). We evaluate the performance of the pattern-matching RE

[5] https://github.com/nelson-liu/lexical-semantic-recognition

[6] Concretely: Locus, Time, EndTime, Goal, Source, Purpose, Duration, Circumstance, ComparisonRef, Manner, Extent.

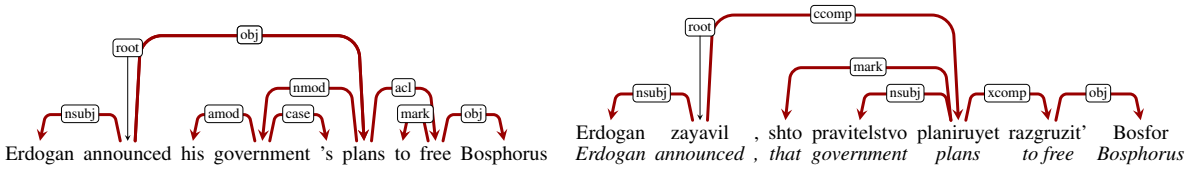

Figure 2: A pair of sentences from the En-Ru PUD corpus that exemplifies a clausal complement (`ccomp` headed by *planiruyet* in the Russian sentence on the right) that is aligned with a nominal complement of the verb of speech (`obj` headed by *plans* in the English sentence on the left).

models trained on the vanilla and transformed versions of the English TACRED dataset. In order to extract syntactic patterns from the test sets, we use a UD parser trained on the vanilla or one of the transformed versions of the English EWT corpus respectively and apply it to the translated sentences in a ZS setting.

When possible, we also report intrinsic evaluation results, with the caveat mentioned above. We compare the performance of a parser trained on the vanilla English EWT corpus against one that is trained on a transformed version of the corpus. As test sets we use our translated and syntactically annotated subset of the TACRED dataset in Russian and Korean (the vanilla version for the first parser and a transformed version for the second parser).

4 Experimental Setup

UD Parser. We use AllenNLP's (Gardner et al., 2018) implementation of the deep biaffine attention graph-based model of Dozat and Manning (2017). We replace the trainable Glove embeddings with the pre-trained multilingual BERT (Devlin et al., 2018) embeddings[7] provided by Hugging Face (Wolf et al., 2020). We also replace the BiLSTM encoder with self-attention to increase the parser's cross-lingual transfer capabilities (Ahmad et al., 2019). Finally, we do not use gold (or any) POS tags to represent a more realistic scenario and avoid introducing the potentially confounding factor of POS divergence into our analysis. We use AllenNLP's default settings and hyper-parameters (all hyper-parameter values are given in Appendix C).

UD Dataset. We use the UD English-EWT corpus for training our models. We use the standard train-dev-test split and v2.5 of the dataset.

Relation Extraction Model. We follow the methodology of Tiktinsky et al. (2020): for each of the representations, we use the TACRED training set (in English) to acquire extraction patterns. We then apply the pattern set to the test sets in Russian and Korean and report F1 scores.

We explore two settings: a STANDARD one, where we remove the source sentences from which we produced the test set in Korean/Russian from the training set, and a simpler PARALLEL one where these examples are included in the training set. PARALLEL emphasizes the ability of the UD parser to produce similar parses in English and in the target language, while STANDARD emphasizes the parser's ability to generalize from other inputs.

Pattern extraction is performed as follows. Given a labeled sentence equipped with a relation name, participant spans, participant types, and a list of trigger words (Yu et al., 2015) collected for different relations (see Appendix B for more details), we find the shortest dependency path between the tokens that connects the entity spans via the trigger words, if such exist.[8] More precisely, if a trigger word is found in the sentence, we first compute the shortest paths between the tokens of the first participant and the trigger word and between the trigger word and the tokens of the second participant. We then form an extraction pattern from the path.[9] If no trigger word is found, we compute the shortest path between the two entities and form a pattern in a similar way.[10] Each pattern is stored in a pattern dictionary together with the number of times it was seen in the training set with a specific relation.

For decoding, we extract the pattern(s) from the input sentence and look up the relations associated with each one in the training pattern dictionary, if such exist. A majority-vote algorithm is then used

[7] Specifically, 'bert-base-multilingual-cased'.

[8] In parser outputs, the entity span sometimes does not constitute a sub-tree. Consequently, there may be multiple paths between entity spans and trigger words, of which we select the shortest one.

[9] For example: `PERSON < nsubj "per_residence" > obj > compound CITY`, where `"per_residence"` is a type of trigger word.

[10] For example: `PERSON < nsubj > obj > compound CITY`.

for prediction.[11]

5 Results and Discussion

Extrinsic Evaluation. The results are presented in Table 2. In the PARALLEL setting, all three transformations display noticeable improvements for both Korean and Russian, increasing both recall and precision. The increase in precision suggests that our transformations not only normalize patterns but also make them more effective. In the STANDARD setting, all three transformations show improvements for Korean. For Russian, however, only the NOMINAL transformation, which lumps several categories into one, is beneficial.

The fact that all three transformations display significant gains in the PARALLEL setting suggests that the UD parser indeed produced similar parses in English and in Russian, as intended. However, the mixed results in the STANDARD setting suggest a difficulty in generalizing across different sentences.

We note that the use of external tools in PREDICATE and OBLIQUE adds a considerable amount of noise to the annotation. For example, TUPA, a parser which is used in the second transformation, obtains an F1 score of only around 70 for the relevant labels. We further note that obtaining improvement for Russian is, on the face of it, more difficult than doing so for Korean as Russian is much more similar to English. Indeed, our analysis in section §2 shows that, in terms of UD, Russian is as close to English as French and the annotation for En-Ru is already very stable.

Therefore, it seems that in the PARALLEL setting the signal for RE is strong enough to combat the added noise, while in the STANDARD setting, which requires generalization across different sentences and thus uses a weaker signal, results are mixed.

To validate our hypothesis, we explore an additional setting that can potentially improve the "signal to noise ratio": we use both the vanilla and the transformed UD parses, for both the RE train and test sets, so that during training and decoding the RE model can use the patterns that appear in either parses. We denote this setting as the ENSEMBLE setting.

The results, reported in Table 3, are an average of all combinations of outputs of (i) one out of ten vanilla-trained parsers and (ii) another vanilla-trained parser (for the baseline) or one out of ten parsers trained on one of the three transformed training sets. We thus mitigate the effect of noisy patterns by allowing the model to take recourse to the vanilla patterns in cases where those are more useful.

We find significant gains in performance for both Korean and Russian with all three transformations. This lends support to our hypothesis that the mixed results obtained in STANDARD for Russian are due to the noise in the implementation of the transformations and thus should not be interpreted as evidence against our hypothesis as to the relation between stability and ZS parsability.

We use the paired bootstrap test to compute the p-values for the the difference between the baseline scores and the different transformations scores. For the ENSEMBLE setting, we find that all positive differences are significant (< 0.05) and the vast majority are highly significant (< 0.001). For the non-ENSEMBLE setting, we find that most are significant (< 0.05).[12]

Intrinsic Evaluation. We report intrinsic evaluation results only for NOMINAL, because the other transformations require parsers for Russian and Korean that are not available to us.

The results, averaged over 10 models, are the following: the vanilla UD parser achieves LAS and UAS of 0.568 and 0.674 for Russian and 0.129 and 0.24 for Korean, respectively. The transformed UD parser achieves the scores of 0.589 LAS and 0.676 UAS for Russian and 0.138 and 0.24 for Korean, respectively. That is, the LAS improvement is of 0.021 and 0.09 LAS points for Russian and Korean, respectively, and is statistically significant.[13] For UAS, the differences are insignificant, which fits in with our discussion in §2, suggesting that the difference in UAS between Partially Aligned and Fully Aligned edges is small.

6 Related Work

The relation between stability and cross-lingual transfer has been the subject of previous work. However, the great majority of it has focused on word order (Wang and Eisner, 2018b; Rasooli and Collins, 2019; Liu et al., 2020a) and morpho-

[11] The pattern extraction and evaluation code is partially based on the pyBART package.

[12] Exact p-values can be found in Appendix E.

[13] We use the paired bootstrap test to compute the p-values for the the difference between the baseline and transformed LAS, all of which are < 0.001.

Trans.	Standard						Parallel					
	Korean			Russian			Korean			Russian		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BASELINE	73.8	12.1	20.8	90.3	18.8	31.1	71.7	12.7	21.6	90.3	18.9	31.3
NOMINAL	77.5	15.7	26.1	90.4	20.4	33.3	75.4	16.6	27.2	90.4	21.6	34.8
	+3.7	**+3.6**	**+5.3**	+0.1	**+1.6**	**+2.2**	**+3.7**	**+3.9**	**+5.6**	+0.1	**+2.7**	**+3.5**
PREDICATE	75.3	12.9	22.1	89.6	18.2	30.2	75.5	13.8	23.3	90.6	21.1	34.2
	+1.5	**+0.8**	**+1.3**	-0.7	-0.6	-0.9	**+3.8**	+1.1	+1.7	+0.3	**+2.2**	**+2.9**
OBLIQUE	77.8	13.1	22.4	90	17.4	29.2	78.6	13.9	23.6	90.6	20.6	33.6
	+4	+1	**+1.6**	-0.3	-1.4	-1.9	**+6.9**	**+1.2**	**+2**	+0.3	**+1.7**	**+2.3**

Table 2: Experimental results for extrinsic evaluation of the transformed versions of UD on the pattern matching RE task across the STANDARD and PARALLEL settings (see §4). Columns correspond to the evaluations settings, target languages, and score types; rows correspond to UD variants. The difference between the scores on the transformed UD corpora and the baseline are shown underlined below the respective scores. Differences with bootstrap p-values < 0.05 are in boldface. The p-values for recall and precision differences for Korean are below 0.08.

Trans.	Standard – Ensemble						Parallel – Ensemble					
	Korean			Russian			Korean			Russian		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BASELINE	71.5	21.6	33	90.1	24	37.9	71.5	22.7	34.3	89	24.2	38.1
NOMINAL	75.6	29	41.8	90.3	27	41.6	74.7	30.6	43.2	90.5	28.8	43.7
	+4.1	+7.4	+8.8	+0.2	+3	+3.7	+3.2	+7.9	+8.9	+1.5	+4.6	+5.6
PREDICATE	71.5	24.7	36.6	88.8	25.1	39.1	72.4	29.6	41.9	89.8	31.1	46.1
	0	+3.1	+3.6	-1.3	+1.1	+1.2	+0.9	+6.9	+7.6	+0.8	+6.9	+8
OBLIQUE	75.9	25.2	37.6	90.7	24.4	38.4	74.2	29.3	41.8	89.9	30.7	45.8
	+4.4	+3.6	+4.6	+0.6	+0.4	+0.5	+2.7	+6.6	+7.5	+0.9	+6.5	+7.7

Table 3: Experimental results for extrinsic evaluation of transformed UD annotations on the pattern matching relation extracting task, compared against the standard UD in the ENSEMBLE setting. Columns and rows are as in Table 2. All positive differences are with bootstrap p-values < 0.05 and the vast majority have p values < 0.001.

logical features (Wang and Eisner, 2018a; Meng et al., 2019) and did not address features that entail stronger structural misalignment. Ponti et al. (2018) and Nikolaev et al. (2020) have shown that cross-lingual divergences in UD annotations of constructions with identical semantics have an effect on cross-lingual transfer but did not precisely quantify this effect. In this work, we proposed a stability-category classification of UD edges and then investigated to what extent the performance of a ZS parser on a given edge is affected by its stability in translation, providing insight into the ability of syntax-based ZS models to generalize over different types of divergences.

Ponti et al. (2018) also demonstrated the benefits of modifying UD parse trees in order to improve their utility for cross-lingual applications. These modifications, however, took UD for granted and only targeted specific types of subtrees to make them look more like the corresponding subtrees in the target language (e.g., by converting an English possessive construction *I have X* into a more Arabic-looking locative-possessive construction *is X at me*). This approach, therefore, is highly language-pair specific as the transformations defined on the differences between surface syntax of English and Arabic will not be useful when presented with another target language. Our work, on the other hand, unconditionally altered the scheme itself, and models using it can be profitably transferred to any target language.

Others (Stanovsky et al., 2014; Schuster and Manning, 2016; Reddy et al., 2017; Nivre et al., 2018; Tiktinsky et al., 2020) also proposed transformations aimed at emphasizing useful connections between tokens but not in a cross-lingual context.

7 Conclusion

Our work establishes a strong relationship between stability and cross-lingual transfer, even at the level of individual edges. We find that the stability of an edge is a strong indicator for the ability of a ZS parser to predict it, suggesting that despite recent advances in ZS parsing and cross-lingual embeddings, these models still face difficulties in generalizing over the grammars and syntactic-usage patterns. Furthermore, we show that it is possible to improve the annotation stability of representations using linguistically motivated and universally applicable transformations, which lead to better

cross-lingual transferability.

Our results suggest several directions for future work in terms of designing more transferable annotation schemes and improving evaluation practices for ZS parsing. They also suggests a path towards a theory of the relation between the linguistic properties of a construction and the ability to effectively process it using cross-lingual-transfer tools.

Acknowledgments

This work was supported by the Israel Science Foundation (grant no. 929/17). Taelin Karidi was partially supported by a fellowship from the Hebew University Center for Interdisciplinary Data Science Research.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Wasi Uddin Ahmad, Zhisong Zhang, Xuezhe Ma, E. Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *NAACL-HLT*.

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

Emanuele Bugliarello and Naoaki Okazaki. 2020. Enhancing machine translation with dependency-aware self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1618–1627, Online. Association for Computational Linguistics.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945, Vancouver, Canada. Association for Computational Linguistics.

Kehai Chen, Rui Wang, M. Utiyama, E. Sumita, and T. Zhao. 2018. Syntax-directed attention for neural machine translation. In *AAAI*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Noelia Ramón García. 2006. Mapping meaning onto form: A corpus-based contrastive study of nominal modification in english and spanish. *Languages in Contrast*, 6(2):307–334.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Multitask parsing across semantic representations. In *Proc. of ACL*, pages 373–385.

Lu Liu, Yi Zhou, Jianhan Xu, Xiaoqing Zheng, Kai-Wei Chang, and Xuanjing Huang. 2020a. Cross-lingual dependency parsing by POS-guided word reordering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2938–2948, Online. Association for Computational Linguistics.

Nelson F. Liu, Daniel Hershcovich, Michael Kranzlein, and Nathan Schneider. 2020b. Lexical semantic recognition. *ArXiv*, abs/2004.15008.

François Maniez. 2012. A corpus-based study of adjectival vs nominal modification in medical english. In Alex Boulton, Shirley Carter-Thomas, and Elizabeth Rowley-Jolivet, editors, *Corpus-Informed Research and Learning in ESP: Issues and Applications*, pages 83–102. John Benjamins.

Marie-Catherine Marneffe, Bill MacCartney, and Christopher Manning. 2006. Generating typed dependency parses from phrase structure parses. volume 6.

Tao Meng, Nanyun Peng, and Kai-Wei Chang. 2019. Target language-aware constrained inference for cross-lingual dependency parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1117–1128, Hong Kong, China. Association for Computational Linguistics.

Amir Pouran Ben Veyseh Minh Van Nguyen, Viet Lai and Thien Huu Nguyen. 2021. Trankit: A lightweight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*.

Phoebe Mulcaire, Jungo Kasai, and Noah A. Smith. 2019. Polyglot contextual representations improve crosslingual transfer. 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 3912–3918, Minneapolis, Minnesota. Association for Computational Linguistics.

Dmitry Nikolaev, Ofir Arviv, Taelin Karidi, Neta Kenneth, Veronika Mitnik, Lilja Maria Saeboe, and Omri Abend. 2020. Fine-grained analysis of cross-linguistic syntactic divergences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1159–1176, Online. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 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, Portorož, Slovenia. European Language Resources Association (ELRA).

Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. 2018. Enhancing Universal Dependency treebanks: A case study. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 102–107, Brussels, Belgium. Association for Computational Linguistics.

Youngbin Noh, Jiyoon Han, Tae Hwan Oh, and Hansaem Kim. 2018. Enhancing universal dependencies for korean. In *Proceedings of the second Workshop on Universal Dependencies (UDW 2018)*, pages 108–116.

Edoardo Maria Ponti, Roi Reichart, Anna Korhonen, and Ivan Vulić. 2018. Isomorphic transfer of syntactic structures in cross-lingual NLP. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1531–1542, Melbourne, Australia. Association for Computational Linguistics.

Ofek Rafaeli, Omri Abend, Leshem Choshen, and Dmitry Nikolaev. 2021. Part of speech and universal dependency effects on english arabic machine translation.

Mohammad Sadegh Rasooli and Michael Collins. 2019. Low-resource syntactic transfer with unsupervised source reordering. 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 3845–3856, Minneapolis, Minnesota. Association for Computational Linguistics.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. 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 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2014. Intermediary semantic representation through proposition structures. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 66–70, Baltimore, MD. Association for Computational Linguistics.

Aryeh Tiktinsky, Yoav Goldberg, and Reut Tsarfaty. 2020. pyBART: Evidence-based syntactic transformations for IE. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 47–55, Online. Association for Computational Linguistics.

Dingquan Wang and Jason Eisner. 2018a. Surface statistics of an unknown language indicate how to parse it. *Transactions of the Association for Computational Linguistics*, 6:667–685.

Dingquan Wang and Jason Eisner. 2018b. Synthetic data made to order: The case of parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1325–1337, Brussels, Belgium. Association for Computational Linguistics.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-lingual BERT transformation for zero-shot dependency parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5721–5727, Hong Kong, China. Association for Computational Linguistics.

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

System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Haoran Xu and Philipp Koehn. 2021. Zero-shot cross-lingual dependency parsing through contextual embedding transformation. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 204–213, Kyiv, Ukraine. Association for Computational Linguistics.

Dian Yu, Heng Ji, Sujian Li, and Chin-Yew Lin. 2015. Why read if you can scan? trigger scoping strategy for biographical fact extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1203–1208, Denver, Colorado. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017a. 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, Vancouver, Canada. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017b. 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, Vancouver, Canada. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

A Additional Data on Edge Stability

In this section, we give additional details on the results reported in the *Edge Stability and ZS Parsability* section (§2).

A.1 Edge Category Percentages and Standard Deviations

Proportions of each edge type in the PUD dataset in percentages are presented in Table 4. Standard deviations for UD zero-shot performance scores for different edge categories are presented in Table 5.

A.2 Edge-Type composition of Different Stability Categories

In order to gain better a insight in the edges in each of the stability categories, we analyze the performance of a supervised parser on these categories. We train 10 supervised models for each language, using the same UD parser as in §4. For Russian, French, Chinese, Korean, and Japanese, we use the GSD corpora. For Arabic, we use the PADT corpus. We use the standard train-dev-test split and v2.5 of the UD dataset.

Results are presented in Table 6, and standard deviations in Table 7. We find that, surprisingly, the parser's performance on edges from different categories is generally ordered in the same order as in the ZS setting. This finding may suggest that cross-lingual stability is correlated with the ability of the parser to generalize within a language as well, a direction which we defer to future work.

We compare the performance difference between the supervised and zero-shot parsers by first normalizing the performance on each category by dividing it by the performances of Fully Aligned edges, thus putting it range from 0 to 1, and then subtracting the normalized score of the supervised parser from the score of zero-shot parser, for each category.

Edge Type	Russian	French	Chinese	Japanese	Korean	Arabic
Fully Aligned	24	23	13	7	11	18
Partially Aligned	7.6	5.3	6.8	3.7	7.8	7.8
Unaligned	20	13	29	45	13	27
Misaligned	5	4	7	4.6	6.5	6
Flipped	1.6	1.3	2.4	1.6	3	2.6
Function Word	42	53	42	38	59	39

Table 4: Edge type proportions in the PUD dataset in percent. Rows corresponds to edge types, columns to PUD dataset languages. Details on the edge types can be found in §2. The values do not sum up to 100% due to rounding.

	Russian		French		Chinese		Japanese		Korean		Arabic	
Edge Type	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Fully Aligned	0.75	0.94	1.02	1.24	1.34	2.15	2.62	2.24	1.36	1.13	1.66	1.77
Partially Aligned	1.61	1.71	1.18	0.82	1.55	1.5	1.91	1.69	1.04	1.63	2.27	1.62
Unaligned	1.4	1.09	0.85	0.9	1.41	1.25	2.77	0.47	1.46	1.15	1.24	1.15
Misaligned	1.13	1.03	1.21	1.21	0.8	0.85	1.22	0.93	1.42	1.04	1.15	0.98
Flipped	1.33	1.71	1.74	1.53	2.04	1.96	1.46	0.89	1.49	0.67	1.36	1.59
Func Word	0.73	0.61	0.41	0.69	1.05	0.71	1.32	1.06	1.67	1.5	0.87	1.02

Table 5: Standard deviations of UD zero-shot performance score per aligned edge type (averaged over 10 models). Rows corresponds to edge types; columns, to evaluation languages and score types. Details on the edge types can be found in §2.

The results are presented in Table 8. We find that the performance difference between the categories is more pronounced in the ZS setting than in the supervised setting. The only noticeable exceptions are in the Partially Aligned and Unaligned edges in Korean and in the Function Word edges in Arabic. We note that the Korean supervised parser displays very poor results, which is most likely due to annotation mismatches between the GSD and PUD Korean corpora.[14] It is also notable that Function-Word edges in Arabic show deviant results in the zero-shot settings as well.

B Translated Resources[15]

Translated RE Datasets In order for the Russian and Korean test sets to be representative of the different relation types in TACRED, we sampled the TACRED training set so that approximately 25% of the examples are labeled as *no_relation* (in TACRED, 79.5% are labeled as such) and the other 75% are proportionally distributed between various relation types. We translated the examples using the Yandex[16] and Papago[17] Translate APIs for Russian and Korean, respectively. Annotators, one for each target language, proficient in the target language and in English, then went over the translated examples, filtering out ones with low-quality translations and sampling others in their stead. They then manually annotated the entity spans of the relation participants corresponding to those identified annotated in English source sentences. The resulting Russian and Korean RE datasets consist of 533 parallel examples in both languages (and 3 additional examples in Russian), thus providing us with parallel, automatically translated, and manually curated and annotated RE datasets.

As the TACRED dataset is annotated with Stanford Dependencies (Marneffe et al., 2006), which are not designed to be cross-lingual, and not with UD, we use the Trankit (Minh Van Nguyen and Nguyen, 2021) supervised parser for parsing the datasets using the default provided pre-trained models. The resulting parses were checked and manually corrected by the annotators.

Translating Trigger words The RE procedure we employ consults a list of trigger words collected for the different relations (Yu et al., 2015).[18] We translate these trigger words to Korean and Russian in two ways. First, we automatically translate the entire word list using an automated machine translating system (Google Translate). This is not always sufficient, as in many cases there are multiple ways to translate a given trigger word. To remedy this, when annotating the translated TA-

[14] Cf. the analysis by Noh et al. (2018).

[15] All resources are available at https://github.com/OfirArviv/translated_tacred.

[16] https://translate.yandex.com/

[17] https://papago.naver.com/

[18] The English trigger-word list is the same as in (Tiktinsky et al., 2020).

Edge Type	Russian		French		Chinese		Japanese		Korean		Arabic	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Fully Aligned	92	89	92	89	85	67	96	95	54	46	82	76
Partially Aligned	92	68	91	73	83	36	97	93	40	19	85	68
Unaligned	87	74	88	81	77	51	93	91	37	26	75	61
Misaligned	74	64	74	68	61	41	85	83	36	27	56	45
Flipped	80	65	83	71	66	50	94	89	36	25	70	48
Func Word	87	83	91	85	68	57	96	95	50	38	72	68

Table 6: UD supervised performance on the PUD corpora per edge type (averaged over 10 models) in percent. Rows corresponds to edge types; columns, to evaluation languages and score types. Details on the edge types can be found in §2.

Edge Type	Russian		French		Chinese		Japanese		Korean		Arabic	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Fully Aligned	0.19	0.22	0.18	0.21	0.37	0.59	0.26	0.18	0.72	0.71	0.25	0.28
Partially Aligned	0.27	0.53	0.24	0.46	0.51	0.45	0.27	0.34	0.57	0.47	0.58	0.51
Unaligned	0.33	0.33	0.19	0.29	0.24	0.22	0.18	0.21	1.54	0.87	0.42	0.31
Misaligned	0.76	0.87	0.75	0.6	0.85	0.69	0.51	0.63	0.91	0.78	0.51	0.67
Flipped	0.74	0.85	0.24	0.53	0.94	1.25	0.38	0.49	0.99	0.81	0.87	0.63
Func Word	0.32	0.33	0.14	0.27	0.21	0.19	0.12	0.16	0.46	0.43	0.36	0.35

Table 7: Standard deviation of UD supervised performance scores per aligned edges type (averaged over 10 models). Rows corresponds to edge types; columns, to evaluation languages and score type. Details on the edge types can be found in §2.

CRED subsample, we also record the spans of the trigger words in the translated sentences that correspond to those in the original English sentences.

C Model Hyperparameters

The hyperparameters of the UD parser are given in Table 9.

D Transformations

D.1 Normalization of Nominal Predicates

One of the prominent sources of cross-lingual discrepancies in translation and when using UD is the handling of nominal predicates, i.e. nouns that evoke a semantic predicate-argument structure, sometimes also called a *scene*. UD does not distinguish between scene-evoking nominal predicates and other nouns; see examples in §3.1.2.

In order to arrive at more cross-lingually stable handling of this construction, we employ UCCA (Abend and Rappoport, 2013), a semantic representation that explicitly distinguishes between scene-evoking and non-scene-evoking nouns. UCCA has two categories for Scene-evoking elements: States and Processes, of which we only consider the latter due to the mediocre parser performance on the former.

We use TUPA (Hershcovich et al., 2018), to parse the training corpus, identify subtrees headed by Processes and convert them to a clause-like form. Specifically, our transformation is as follows:

- `nsubj` (nominal subject) is converted to `csubj` (clausal subject).
- `nmod` (nominal modifier) is converted to `acl` (adnominal clause).
- `compound` is similar to `nmod` and thus is converted to `acl` as well.
- `obj` and `iobj` are converted to `ccomp` (complement clause).
- `obl` signify both indirect objects (*Take money from a stranger*) and adverbial relations (*Come back before tomorrow*) and therefore correspond to either `advcl` (adverbial clause) or `ccomp`. We convert them to `advcl` as adverbial-clause semantics in `obl` is more common, according to Nikolaev et al. (2020).

As we change the labels of nodes from nominal types to clausal types, we need to also change the labels of their dependents because, per UD convention, clause heads cannot be modified by adnominal dependents. We convert those nodes to their closest clause-level equivalents. We adopted the following approach:

- Adjectives and adverbs are often very close semantically (*his play is magnificent* $\rightarrow$ *he plays magnificently*), hence we convert `amod` to `advmod`.

Edge Type	Russian		French		Chinese		Japanese		Korean		Arabic	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Fully Aligned	0	0	0	0	0	0	0	0	0	0	0	0
Partially Aligned	0.07	0.26	0.04	0.14	0.05	-0.07	-0.01	0.3	-0.23	-0.18	0.11	0.45
Unaligned	0.12	0.12	0.07	0.09	0.18	0.19	0.39	0.69	-0.07	-0.06	0.14	0.16
Misaligned	0.19	0.17	0.17	0.17	0.26	0.27	0.47	0.58	0.04	0.23	0.1	0.1
Flipped	0.33	0.37	0.29	0.26	0.34	0.47	0.61	0.66	-0.01	0.23	0.24	0.17
Func Word	0.18	0.18	0.08	0.07	0.21	0.34	0.61	0.65	0.26	0.31	-0.09	-0.26

Table 8: Comparison of the performance between the supervised and zero-shot parsers on each edge category. We first normalize the performance of each category by dividing it by the performances of Fully Aligned edges and then subtract the normalized score of the supervised parser from the score of zero-shot parser, for each category.

Input	Input dropout rate: 0.3 Token embedder: bert-base-multilingual-cased
Encoder	Type: Stacked self attention Input dim: 768 Hidden dim: 400 Projection dim: 512 feedforward hidden dim: 400 Layers #: 3 Attention heads #: 8
MLP and Attention	Arc MLP size: 500 Label MLP size: 100 MLP layers #: 1 Activation: ReLU Dropout: 0.3
Training	Batch size: 128 Epochs #: 100 Early stopping: 50 Adam lrate: 0.001 Adam $\beta 1$: 0.9 Adam $\beta 2$: 0.9

Table 9: Hyper-parameters used in the deep biaffine attention parser used in our experiments.

- `acl` is converted to `advcl`.

- `nmod` and `compound` represent participants of a scene denoted by a nominal predicate. In a clause, a participant can be `nsubj`, `obj`, `iobj` or `obl` (cf. *killing **of an artist***: was the artist killed [obj] or did they kill somebody [nsubj]?). Choosing the correct participant type is a hard semantic task that we do not have good instruments to solve. Instead we introduce a new label, `A`, an undifferentiated participant, and convert `nmod` and `compound` to it.

The normalization of `nmod` and `compounds` into `A` creates a discrepancy: clauses transformed by our transformation contain `A`, while those were not still contain `nsubj`, `obj`, `iobj`, and `obl`. To harmonize this discrepancy we convert `nsubj`, `obj`, `iobj`, `obl` in all clauses into `A` as well. This results in a considerably simplified version of UD.

E Extrinsic Evaluation p-values

We use the paired bootstrap test to compute the p-values for the the differences between baseline scores and different transformation-based scores. For the ENSEMBLE setting, we find that all positive differences are significant (< 0.05) and the vast majority are highly significant (< 0.001). For the non-ENSEMBLE setting, we find that most are significant (< 0.05). p-values are presented in Tables 10 and 11.

	Standard						Parallel					
	Korean			Russian			Korean			Russian		
Trans.	P	R	F1	P	R	F1	P	R	F1	P	R	F1
NOMINAL	0.000281	0.000235	0.000164	0.442	0.0153	0.0167	0.000303	0.00022	0.000213	0.415	0.000211	0.000181
PREDICATE	0.193	0.0459	0.0483	0.878	0.954	0.953	0.0422	0.0808	0.0725	0.323	0.000108	0.000118
OBLIQUE	0.00164	0.0566	0.05	0.69	1	1	3.83e-05	0.0062	0.00225	0.345	1.54e-05	9.72e-06

Table 10: p-values, computed using the paired bootstrap test, for the extrinsic evaluation of a transformed UD annotation on the pattern matching RE task in the STANDARD and PARALLEL settings. Columns and rows are the same as in Table 2.

	Standard–Ensemble						Parallel–Ensemble					
	Korean			Russian			Korean			Russian		
Trans.	P	R	F1	P	R	F1	P	R	F1	P	R	F1
NOMINAL	7.47e-05	5.6e-05	4.01e-05	0.00262	5.34e-05	4.66e-05	0.000114	4.38e-05	6.21e-05	8.47e-05	6.13e-05	5.62e-05
PREDICATE	0.522	4.14e-05	4.9e-05	1	4.49e-05	4.03e-05	5.55e-05	5.46e-05	6.41e-05	8.35e-05	4.65e-05	3.43e-05
OBLIQUE	4.74e-05	4.88e-05	4.86e-05	5.35e-05	0.037	0.0188	4.02e-05	4.09e-05	5.03e-05	0.000197	3.57e-05	3.29e-05

Table 11: p-values, computed using the paired bootstrap test, extrinsic evaluation of a transformed UD annotation, using the pattern matching RE task, compared against the standard UD, for both Russian and Korean in the ENSEMBLE settings. Columns and rows are as in Table 2.