Assessing the Role of Lexical Semantics in Cross-lingual Transfer through Controlled Manipulations

Anonymous ACL submission

Abstract

While cross-linguistic model transfer is effective in many settings, there is still limited understanding of the conditions under which it works. In this paper, we focus on assessing the role of lexical semantics in cross-lingual transfer, as we compare its impact to that of other language properties. Examining each language property individually, we systematically analyze how differences between English and a target language influence the capacity to align the language with an English pretrained representation space. We do so by artificially manipulating the English sentences in ways that mimic specific characteristics of the target language, and reporting the effect of each manipulation on the quality of alignment with the representation space. We show that while properties such as the script or word order only have a limited impact on alignment quality, the degree of lexical matching between the two languages, which we define using a measure of translation entropy, greatly affects it.

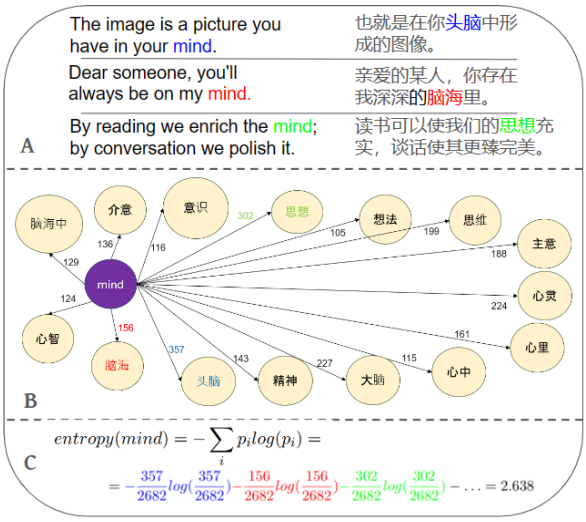

Figure 1: **A.** Sentences from the *UM* parallel corpus. In each sentence, the word *mind* is colored along with its translation in Simplified Chinese. **B**. A weighted graph which results from the UM corpus. The edge weights indicate how many times *mind* is translated into each instance in Simplified Chinese. **C.** Calculation of the *translation entropy* of the word *mind* in the UM corpus.

1 Introduction

Different languages partition meanings over their vocabularies in different ways. In English, the concept *wall* includes both a structural component in a house and a defensive barrier around a city, whereas Spanish distinguishes them with the concepts *pared* and *muro*. This raises the question of how such differences in lexical semantics influence cross-lingual transfer – the ability of models trained on data from one language to effectively perform tasks in another language (Kim et al., 2017; Artetxe and Schwenk, 2019a; Dobler and de Melo, 2023).

In this work, we study the impact of lexical semantics and other linguistic properties on the effectiveness of cross-lingual transfer. We examine how various properties affect the ability to extend an existing representation space to include an additional low-resource language, and consequently, how they affect the zero-shot performance of the low-resource language.

To isolate the distinct linguistic properties and evaluate their individual impact, we perform manipulations to the English language that mimic specific language traits found in other languages, thereby creating artificial languages. For instance, to evaluate the impact of lexical semantics, we create an artificial language by imposing lexicalization patterns of other languages onto English.

We define a weighted bipartite graph that links the vocabularies of two languages, mapping each word in one language to all its potential translations in the other language. We leverage this graph to characterize the lexicalization patterns between the languages in information theoretic terms.

Our results indicate that the lexicalization patterns of the source and target languages have more

impact on transferability than other linguistic properties. They also demonstrate robust correlation between the entropy of words in the bipartite graph we define and zero-shot performance.

2 Related Work

2.1 Cross-lingual Transfer Methods

Multilingual language models (MLLMs) like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020a) exhibit remarkable zero-shot cross-lingual performance, despite being trained without parallel data. However, they also face limitations. Being contextualized token embeddings, they may underperform in sentence-level tasks (Hu et al., 2020b). Moreover, training these models requires a massive amount of text from each language, posing a major challenge to the inclusion of low-resource languages.

To overcome these limitations, Reimers and Gurevych (2019, 2020) trained a model (Sentence-BERT) using a sentence-level objective to obtain sentence representations (2019). They then employed knowledge distillation (teacher-student supervised learning) to extend the representation space to additional languages (2020). This approach, while requiring parallel data, proves effective with relatively few samples, making it suitable for low-resource languages. Heffernan et al. (2022) applied a similar technique with LASER2 (Artetxe and Schwenk, 2019b), a language-agnostic sentence encoder, as their teacher model. They demonstrated the efficiency of this approach with extremely low-resource languages. In our research, we follow a similar setup.

2.2 Investigations of Zero-shot Transfer

Many studies examined the factors affecting zero-shot cross-lingual transfer. Some defined metrics of language similarity, such as geographical, genetic, or phonological distance, and explored their relation to transferability (Lin et al., 2019; Lauscher et al., 2020; Dolicki and Spanakis, 2021; Ahuja et al., 2022). Others focused on assessing the impact of specific properties like lexical overlap (Wu and Dredze, 2019; Patil et al., 2022; de Vries et al., 2022) or syntactic structure and word order (Dufter and Schütze, 2020; Arviv et al., 2021; de Vries et al., 2022; Chai et al., 2022; Wu et al., 2024).

Among the studies exploring word order, some rearranged English sentences to create artificial languages and analyzed their behavior (Dufter and Schütze, 2020; Deshpande et al., 2022; Chai et al., 2022; Wu et al., 2024). As summarized by Philippy et al. (2023), experiments involving sentence inversion or random shuffle showed a significant decline in zero-shot performance compared to experiments where the word order was systematically modified based on the structure of other natural languages. This implies that while word order is important, differences in word order between languages may play a minor role in zero-shot transfer.

To our knowledge, no previous work examined the impact of variations in the lexicalization patterns of the two languages on cross-lingual transfer learning. In this paper, we aim to address this gap. In addition, while previous studies mostly focused on representations derived from masked language models, we focus on a knowledge distillation setup.

2.3 Lexicalization Patterns

Lexicalization patterns were widely used in linguistic typology to classify languages and explore language universals, in cognitive science to study conceptualisation, and even by anthropologists to examine cultural influences on language and cognition (François, 2008; Jackson et al., 2019; Xu et al., 2020; Georgakopoulos et al., 2022). The majority of research on lexicalization has been centered around the concept of colexification (a linguistic phenomenon that occurs when multiple concepts are expressed in a language with the same word). Traditionally, colexification data relied on hand-curated resources, but this changed with the introduction of CLICS (List et al., 2018), promoting exploration into large-scale colexification graphs also in NLP (Harvill et al., 2022; Liu et al., 2023a,b).

Liu et al. (2023b) proposed a more systematic way to investigate the conceptual relation between languages and extract colexifications. Their method includes aligning concepts in a parallel corpus and extracting a bipartite directed graph for each language pair, mapping source language concepts to sets of target language strings. Leveraging these bipartite graphs, they identify colexifications across a diverse set of languages. Here, we employ a similar method, albeit to a different purpose – our primary focus lies in proposing a methology for quantifying how differences in lexical semantics impact cross-lingual transfer.

3 Method

Our main goal in this paper is to study how different language properties, with a particular emphasis on

lexical patterns, influence the ability to perform cross-lingual transfer, and we aim to do so in a carefully controlled way.

To isolate distinct language properties and understand their respective contribution, we define different manipulations of the source language L_s. For each of these manipulations, we modify L_s so that it imitates certain properties found in a target language L_t, creating a new artificial language $L_\mathcal{A}$ (Section §4). Throughout this article, we maintain English as the source language.

Then, for each artificial language $L_\mathcal{A}$, we follow the distillation method proposed by Reimers and Gurevych (2020), training a model to encode sentences of $L_\mathcal{A}$ into an English pretrained representation space. We explore which of them allows for an effective knowledge transfer and, hence, performs well in a zero-shot setting (Section §6).

Model Distillation. The pretrained teacher model we use is an English sentence transformer model (Reimers and Gurevych, 2019). It is trained using English sentence pairs and a self-supervised contrastive learning objective to encode similar English sentences into vectors that are close to one another in the vector space. Given a sentence from a pair, the model is trained to predict which of a batch of randomly sampled other sentences is in fact paired with it. The outcome of this training yields a sentence representation space that captures the semantic information of a given sentence. Within this pretrained vector space, the cosine similarity between two vectors indicates the degree of similarity between the two sentences they represent.

The pretrained representations of the teacher model serve us throughout the experiment as ground truth. For each artificial language $L_\mathcal{A}$, we train smaller transformer models using an *English*–$L_\mathcal{A}$ parallel corpus. Denoting the teacher model with M and the student model that corresponds to the language $L_\mathcal{A}$ with $m_\mathcal{A}$, for each sentence pair $(s, t) \in$ *English*$\times L_\mathcal{A}$, the training objective is to minimize the following cosine embedding loss:

$$L_{cos}(m_\mathcal{A}(t), M(s)) = \begin{cases} \mathbf{cos}(\mathbf{m}_\mathcal{A}(\mathbf{t}), \mathbf{M}(\mathbf{s})) \\ \qquad \text{if } t \text{ is a manipulation of } s \\ \mathbf{max}(\mathbf{0}, \mathbf{cos}(\mathbf{m}_\mathcal{A}(\mathbf{t}), \mathbf{M}(\mathbf{s}))) \\ \qquad \text{otherwise} \end{cases} \quad (1)$$

This optimization process aims to increase the cosine similarity in the vector space whenever the sentence t is a manipulation of the sentence s, and decrease it in any other case. As a result, it produces a sentence encoder that maps each sentence $t \in L_\mathcal{A}$ to a location in the pretrained vector space as close as possible to the representation of the original English sentence.

Evaluation. For each student model $m_\mathcal{A}$, we compute the average cosine similarity between the embeddings of English sentences and the embeddings of the corresponding manipulated sentences within a held-out subset of the corpus. This serves as the intrinsic evaluation. Additionally, we employ the model in a zero-shot experiment for an extrinsic NLP task and present its performance. These two outcomes help us understand the quality of the alignment of the language $L_\mathcal{A}$ with the pretrained representation space of the teacher model.

4 Manipulations of the Data

We proceed to define the different manipulations that we apply. For each manipulation, we modify the English source to create an artificial language $L_\mathcal{A}$, generate an *English*–$L_\mathcal{A}$ parallel corpus, and train student models $m_\mathcal{A}$ as explained.

Our primary focus lies within the domain of lexical semantics. To thoroughly examine their influence, we take a broader approach, investigating how the effect of lexical semantic manipulations compare with that of other linguistic properties. We examine three aspects: script, syntax, and lexical semantics. For each of these aspects, we define a manipulation that solely modifies it. First, we substitute the letters of the English alphabet with symbols of a different script (§4.1). Second, we systematically rearrange the word order in sentences, thus examining the effect of the syntactic structure, or at least a specific aspect of it (§4.2). Finally, we replace the English lexicon with that of a target language to explore the significance of variations in lexicalization patterns (§4.3). By isolating each linguistic aspect, we obtain a clearer understanding of its individual contribution.

4.1 Manipulating the Script

To manipulate the script we simply substitute each English character with a symbol from another script in an injective manner. For instance, if we consider the Greek alphabet system, we can swap the characters according to their sequential order: $a \to \alpha$, $b \to \beta$, $c \to \gamma$, and so forth. This way, the sentence *Brown cows eat grass* will transform into: $\beta\sigma\sigma\psi\xi$ $\gamma\sigma\psi\tau$ $\epsilon\alpha\upsilon$ $\eta\sigma\alpha\tau\tau$.

4.2 Manipulating Word Order

The second manipulation we use is a word reordering one. We apply the word reordering algorithm developed by Arviv et al. (2023), to permute the words of each source sentences so that it will conform to the syntactic structure of the target language (see Appendix B for full details). The algorithm recursively reorders all the subsequences in a source sentence, yielding a new sentence in an artificial language $L_{\mathcal{A}}$ that imitates the word-order of a target language L_t. For example, the sentence *Brown cows eat grass* yields different results depending on the selected target language. Spanish, an *SVO* language, but in which nouns are ordered before adjectives, produces: *Cows brown eat grass*; whereas Hindi, an *SOV* language, produces: *Brown cows grass eat*.[1]

4.3 Manipulating the Lexicon

The core of our study is lexical semantics and their impact on cross-lingual transfer. We seek to assess the influence of the diverse distribution of meanings across different lexicons. To achieve this, we develop a manipulation in which we substitute the lexicon of the source language L_s with that of a target language L_t. This creates a new artificial language $L_{\mathcal{A}}$ that is based on the lexicon of L_t while retaining the original sentence structure of L_s.

The manipulation is based on a word alignment between a source sentence $s \in L_s$ and its translation in the target language $t \in L_t$. We replace each word in the source language with its corresponding translation in the target language, thus adopting the lexical semantics of the target language while preserving the original syntax.[2]

However, word-aligned bitext is difficult to obtain. While manually aligned parallel datasets are scarce and limited in size, model-based automatic aligners often map words to a wide range of possible translations. When defining a lexical manipulation that depends on mapping one lexicon to another, ensuring each word consistently maps to the same set of words is crucial. To address this, we develop an algorithm that refines the output of an automatic aligner, mapping each word of the lexicon to a fixed set of words. This careful process involves extracting a bipartite graph from the bitext, which ensures consistent mapping.

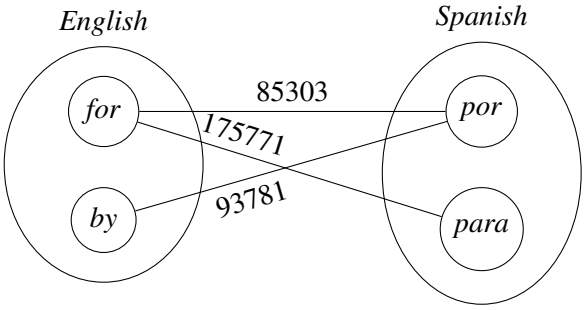

Figure 2: Illustration of the weighted sub-graph which results from the *Europarl* parallel corpus. The edges represent the possibility that two words are translations of each other. The weights denote the number of occurrences that each word pair is aligned in the bitext.

Formalism. Consider a word-aligned bitext that contains the languages L_s and L_t. We define $G = (V_s, V_t, E, w)$ to be a weighted bipartite graph, where V_s is the set of words in the lexicon of L_s, and V_t is the set of words in the lexicon of L_t. A pair of words $(v, u) \in V_s \times V_t$ is an edge in G iff v is aligned to u in at least one instance in the bitext. The weight function $w : E \to N^+$ assigns the number of times that each word pair is aligned in the bitext.

This construction aims to capture the relationship between the lexical semantics of two languages. For example, the Spanish translation of *for* is *por* in some cases and *para* in others; *by* is also occasionally translated as *por* (e.g., *multiply by three* translates to *multiplicar por tres*). We thereby obtain the subgraph in Figure 2.

We hypothesize a negative correlation between the degree of the vertices in the graph and the ability to perform cross-lingual transfer between the languages. In other words, the closer the lexicons approximate a bijective relationship, the better we expect the cross-lingual transfer performance to be.

Swapping Algorithm. We proceed to outline the systematic procedure we employ to perform the lexical manipulation. For each pair of languages L_s, L_t we follow these steps:

1. We apply an automatic word aligner to a large L_s–L_t parallel corpus, extracting a weighted bipartite graph G as described above.

[1] Since the algorithm is based on fixed statistics, the artificial language it produces exhibits a more consistent word order than that of a natural language. We prefer this experiment over one in which the order of words in a sentence is randomly rearranged due to the potential noise this might add.

[2] This manipulation inherently includes the first manipulation, at least to some extent, as altering specific words in the sentence also influences the script. However, we will demonstrate later that the script is not a significant factor, making this fact of minor importance to our conclusions.

2. We filter out of the graph any edge $e \in E$ that represents an alignment which is not substantial (that is, an edge whose weight does not exceed a certain threshold or whose weight is relatively small compared to other edges originating from the same vertex).[3]

3. Given a source sentence $s \in L_s$ and its translation in the target language $t \in L_t$, we run the automatic aligner to achieve a word-to-word alignment between s and t.

4. For each source word $v \in s$:

(a) If there exists a word $u \in t$ such that the word-to-word alignment includes the pair (v, u), and at the same time it holds that $(v, u) \in E$, then we replace the word v with u.

(b) Otherwise, if there exists a word $u \in t$ such that $(v, u) \in E$, we replace the words as well. If there is more than one valid choice, we select the word $u \in t$ for which the weight $w(v, u)$ is the highest.

(c) Otherwise, we look for the edge $(v, u) \in E$ that has the highest weight among all edges originating from v, meaning that u is the most common alignment of v in the language L_t. If we find such an edge, we replace v with u.[4]

(d) In a case there are no edges originating from v, we preserve it.

This systematic procedure provides a mapping between two lexicons, and therefore enables us to make consistent decisions for each word in the lexicon – whether to be replaced or preserved. This helps maintaining a coherent semantic structure in the resulting artificial language $L_{\mathcal{A}}$.[5]

For a simple illustration, consider the sentence *Brown cows eat grass* and its Spanish translation *Las vacas marrones comen hierba*, and assume the auto-aligner's output is *brown→marrones*, *eat→comen*. When applying our swapping algorithm, we first check whether the edges *(brown–marrón)* and *(eat–comer)*[6] appear in the bipartite graph. As both of them do, we replace *brown* with *marrones* and *eat* with *comen*. Next, we search for words in the target sentence that are linked to the source words in the graph, resulting in the edges *(cow–vaca)* and *(grass–hierba)*. These two words are swapped with their corresponding pairs as well. This process ultimately yields the sentence: *Marrones vacas comen hierba*.

Translation Entropy. To further appreciate the impact of the divergence between the source and the target lexicons, we introduce the concept of *translation entropy*. Let G be the weighted bipartite graph presented earlier, we compute the entropy for each vertex v in the graph:[7]

$$e(v) = -\sum_{u \in U_v} p_v(u) log(p_v(u)) \qquad (2)$$

where U_v is the subset of vertices linked to v, and p_v is the following probability function:

$$p_v(u) = \frac{w(v, u)}{\sum_{u' \in U_v} w(v, u')} \qquad (3)$$

As w counts occurrences of each word pair aligned in the bitext, the outcome of the function p_v is the probability that, in a particular instance, the word v is linked to the word u among all possible $u' \in U_v$ (see calculation example in Figure 1).

We examine the impact of *translation entropy* in two distinct configurations: one for the source words (Figure 3A) and another for the target words (Figure 3B). In the first, we compute the *translation entropy* for all source vertices $v \in V_s$ and partition the set V_s into three disjoint subsets based on the percentile of the *translation entropy* values. The percentile calculation is based on the number of instances in the database, rather than the number of words in the lexicon. Then, in each experiment, we remove from the graph G all the source vertices that do not belong to a specific subset to achieve the sub-graph G'. We apply once again the lexical manipulation, but this time using the filtered graph G'. In the second configuration, we follow the exact same steps, but this time for the target vertices $v \in V_t$.[8]

[3] These parameters may depend on the target language. See Appendix C.

[4] In languages where the words have different inflections, we check the validity of the match based on the lemma, but replace the words in their original form. The determination of the most common alignment also considers the original inflection.

[5] To ensure that our algorithm does not harm the alignments of the auto-aligner, we compare the precision and recall of our algorithm's outputs with the original alignments of the auto-aligner against the gold standard. We observe higher precision but lower recall, resulting in a slightly better F1 score for our algorithm's outputs. For further details, please refer to Appendix D.

[6] The lemmas of *marrones* and *comen* respectively.

[7] It does not matter whether $v \in L_s$ or $v \in L_t$.

[8] The two configurations are not directly comparable: re-

Returning to the previous example, after we extract an *English–Spanish* bipartite graph from the *Europarl* parallel corpus and compute the entropy of the English words, we obtain: *e(brown)=0.545*, *e(cow)=0.24*, *e(eat)=0.631*, *e(grass)=0.799*. If we filter the graph to retain only words that fall within the upper third of *translation entropy* values, we find that the word *grass* is the only one meeting this criterion. Consequently, applying our lexical manipulation to this filtered graph results in the sentence: *Brown cows eat hierba.*

5 Experimental Setup

5.1 Datasets

In this subsection, we outline the datasets used in our work. For further details on the datasets, their selection rationale, and the filtering process applied, please refer to Appendix E. The bitext we use for training and intrinsic evaluation of the student models throughout all manipulation experiments is the *TED-2020* parallel corpus (Reimers and Gurevych, 2020). To avoid over-specializing the tokenizer on this small dataset, we train the tokenizers on the *CC-100* corpus (Wenzek et al., 2020).[9] The CC-100 corpus is also used for our monolingual benchmark experiments (see beggining of Section §6). For extrinsic evaluation we use the *Cross-lingual Natural Language Inference (XNLI; Conneau et al., 2018)*. To extract the bipartite graphs, we require a large parallal corpus. Therefore, we use the *Europarl* corpus for Europian languages and the *UM* corpus (Tian et al., 2014) for Simplified Chinese.

5.2 Models

Teacher Model. We select the pretrained sentence transformer *all-mpnet-base-v2*. This model was trained on 1B English sentence pairs with a self-supervised contrastive learning objective (see Section §3). The training produced a 768-dimensional vector space that has proven to achieve state-of-the-art results in sentence-level tasks.

Student Models. We train multiple RoBERTa models (Liu et al., 2019), with each model designed to encode a sentence into the teachers' 768-dimensional vector space. To achieve this, we add a mean-pooling layer on top of the last hidden layer. We set the vocabulary size to 30527, matching that of the teacher model, the number of max position embeddings to 28, and the hidden size to 768. As to the number of hidden layers and the number of attention heads, we explore various architectures: 3/6/9/12 hidden layers paired with 4/6/8/12 attention heads, respectively. The number of trainable parameters for these configurations are 84316, 135004, 185692, and 236380, respectively. We reserve a small portion of the dataset for testing (20K sentence pairs in the *TED* corpus and 100K sentences in *CC-100*), and then randomly split the training set into 90% for actual training and 10% for validation. We use the Adam optimizer with a learning rate of $3e^{-5}$, continuing until the validation loss does not decrease for five consecutive epochs. The model with the lowest validation loss is selected, and its performance on the test set is reported.

NLI Model. For the zero-shot English NLI experiment, we train a Multi-Layer Perceptron (MLP) on top of the teacher model. We use the usual combination of the two sentence embeddings: $(p; h; p \cdot h; |p - h|)$, where p and h are the premise and the hypothesis respectively (see for example Artetxe and Schwenk, 2019b). We build the MLP with two hidden layers of size 128, and train it for 150 epochs using the Adam optimizer. We select the model that achieves the lowest loss on the test set.

Auto-aligner. For obtaining high-quality word-to-word alignments we use the *Simalign* automatic aligner (Jalili Sabet et al., 2020). This tool uses contextualized embeddings to map words between sentences. We run it with *XLM-R* as the base model, and set the matching method to be $ArgMax$. To simplify the analysis, we filter its outputs to include only one-to-one alignments. We manually review some of the alignments to ensure their quality.

6 Experiments & Results

To assess the impact of each linguistic property on transferability, we apply our manipulations to English and carry out the distillation process for each artificial language $L_{\mathcal{A}}$. For intrinsic evaluation, we compute the average cosine similarity (hereafter: ACS) between the embeddings of English sentences and the embeddings of the corresponding manipulated sentences in the test set. For extrinsic evaluation, we use XNLI zero-shot accuracy.

Before manipulating English, we conduct experiments to obtain reference points for evaluating the

moving a source word from the lexicon leads to a reduction in the number of swaps performed, whereas removing a target word reduces the diversity of the swaps but not necessarily the number of them.

[9] Full details on tokenizer training for each artificial language can be found in Appendix F.

models. First, we perform the distillation process on regular English sentences from the CC100 corpus. We explore the influence of varying training set sizes and of the student model architecture. We observe a considerable margin, with differences of up to 0.227 in ACS, between models trained on 50K sentences and those trained on 1M. Conversely, we observe a smaller margin, with differences of up to 0.035 in ACS, between smaller and larger model architectures. Our findings suggest that for low-resource scenarios, exceeding 6 hidden layers and 6 attention heads is unnecessary. For full results refer to Appendix A.1.

Second, we conduct cross-lingual experiments using the *TED-2020* parallel corpus to compare English with other natural languages without manipulation. The results of the cross-lingual experiments serve as a lower bound for the performance on the manipulated data (as the manipulations are meant to change English to be closer to the target language). We also train student models on English with a newly trained tokenizer to arrive at an upper bound. We observe a substantial range between the lower and upper bounds, with differences of up to 0.186 in ACS, which gives us sufficient room to experiment with our manipulations. See full results in Appendix A.2.

We proceed to apply our manipulations to tease apart the properties of the data that contribute to this difference between in-language training and cross-lingual training.

6.1 Script Substitution

We perform two script substitutions: first, replacing the English characters with Greek characters sorted alphabetically, and second, replacing them with Simplified Chinese characters sorted by their frequency. For all the student models, we train a tokenizer from scratch (see Appendix F for full details). The results, reported in Table 1, show almost no degradation in performance.

	ACS score		XNLI accuracy	
	50K	**100K**	**50K**	**100K**
English	0.725	0.786	55.7	59.4
Greek alphabet	-0.0	-0.0	-1.1	-0.7
Chinese symbols	+0.003	+0.002	-0.4	-1.0

Table 1: Results for the script substitution experiment. 50K/100K denote for the number of training sentences.

6.2 Word Reordering

We apply the reordering algorithm developed by Arviv et al. (2023) each time relying on the *pairwise ordering distributions* of a different language. We examine *SVO* languages (Spanish, Greek, Chinese and Hebrew) as well as an *SOV* language (Hindi). Results are presented in Table 2. Although we observe a degradation in performance, it is a very slight one. The ACS score in the worst case (100K Greek sentences) decreases by 0.013 points, and the XNLI accuracy in the worst case (100K Hindi sentences) decreases by 1.5%.

	ACS score		XNLI accuracy	
	50K	**100K**	**50K**	**100K**
English	0.725	0.786	55.7	59.4
Spanish order	-0.002	-0.007	+0.2	-0.5
Greek order	-0.007	-0.013	-0.9	-1.3
Chinese order	-0.002	-0.012	-0.6	-1.3
Hebrew order	-0.0	-0.005	-0.5	-0.4
Hindi order	-0.005	-0.01	+0.8	-1.5

Table 2: Results from the distillation process for the word reordering experiment.

6.3 Lexical Swapping

We follow the steps described in Section §4.3 for Spanish, Greek and Simplified Chinese. When constructing the weighted bipartite graph, for Spanish and Greek we use the datasets *Europarl+TED*, whereas for Simplified Chinese we use *UM+TED*. Results are presented in Table 3. In this experiment, we observe a significant decrease in both the ACS score and the XNLI accuracy. The language that performs the worst is Simplified Chinese, with up to 0.092 degradation in the ACS score and up to 5.9% in XNLI accuracy.

	ACS score		XNLI accuracy	
	50K	**100K**	**50K**	**100K**
English	0.725	0.786	55.7	59.4
Spanish lexicon	-0.055	-0.06	-2.3	-1.9
Greek lexicon	-0.073	-0.073	-3.4	-3.3
Chinese lexicon	-0.079	-0.092	-4.8	-5.9

Table 3: Results from the distillation process for the lexical swapping experiment.

These results suggest that variations in lexicons significantly impact the capacity to align a language with a pretrained representation space, thereby affecting transferability. To gain a deeper understanding of this phenomenon, we proceed to applying the same manipulation, this time selectively swapping only a subset of the words in the language.

Entropy-based Lexical swapping. In this experiment we filter the vertices of the bipartite graph based on their *translation entropy* (see §4.3) and

then apply the lexical swapping manipulation. Figure 3A presents the outcome of filtering the source vertices, and Figure 3B shows the result of filtering the target vertices. In both cases, we split the set of vertices based on percentiles: into the ranges of 0-33, 33-67, and 67-100. In addition, we include an experiment where we exclusively swap words with zero entropy, and we add the results from the full lexical manipulation.

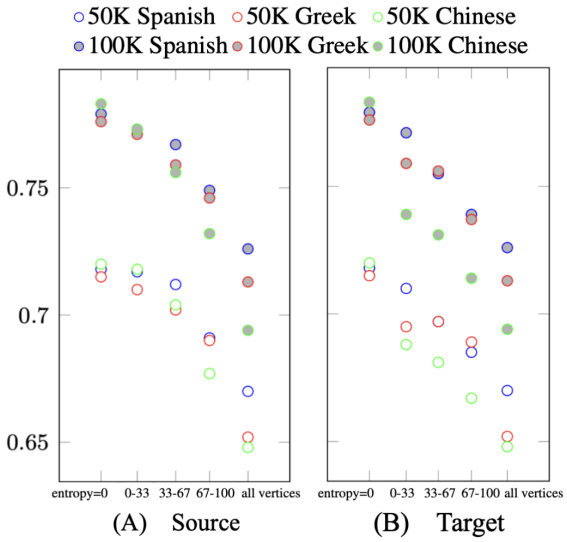

Figure 3: Results of filtering the **source** vertices in the graph in Figure (A) and results of filtering the **target** vertices in the graph in Figure (B). The horizontal axis represents the entropy values ranging from 0 to *all vertices*. Numerical x values denote percentiles. The y axis represents ACS scores.

We observe a robust negative correlation between the entropy of the words we swap and the similarity scores. In all cases except for one (filtering Greek words of percentile 33-67), the higher the entropy of the words swapped, the worse the distillation process performs. Moreover, when we swap only 33% of English word instances with low entropy, it has minimal impact on performance, but when we swap 33% of word instances with the highest entropy, it results in a degradation of performance that is close to the degradation observed in the full lexical manipulation. We conclude that swapping in itself does not degrade performance; instead, most degradation results from the lexicons not being aligned in a one-to-one manner.

The absence of one-to-one alignment in the lexicons conceals two separate phenomena: synonymy and polysemy. In case of a synonymy, a specific word is translated to different words in different contexts, whereas in the case of polysemy, several distinct words are translated to the same word. The first experiment (filtering the source words) mostly simulates the impact of synonymy, while the second experiment (filtering the target words) mostly simulates the impact of polysemy. Results imply that both phenomena have a substantial impact on cross-lingual transfer.

7 Conclusion

We leverage a knowledge distillation setup to explore the conditions that allow successful cross-lingual transfer. We apply various manipulations to English to alter specific language properties and assess their impact.

First, we apply script substitution and observe almost no degradation in performance. Next, we examine the impact of word order. Unlike previous studies (Deshpande et al., 2022; Chai et al., 2022; Wu et al., 2024), which made only subtle modifications to the constituent order in some experiments and inverted/shuffled all the words in others, we apply a manipulation that permutes many words in the sentence but still maintains a coherent syntactic structure. We believe that this manipulation provides us with a more nuanced understanding of how word order affects cross-lingual transfer. Our findings align with the survey by Philippy et al. (2023), suggesting that as long as the syntactic structure remains coherent, the effect of word order is less substantial (compared to inversion/shuffling).

Finally, we swap words from the English lexicon with words from the target lexicon and observe a substantial degradation in performance. We use the notion of *translation entropy* to explore the impact of swapping only a subset of words in the lexicon. This reveals that swapping the words with the highest entropy leads to a more substantial degradation in performance compared to words with lower entropy. These findings support our hypothesis: the more the lexicons align in a one-to-one manner, the better cross-lingual transfer will perform.

To recap, Among the three manipulations we apply then, only lexical swapping was found to have a substantial effect. This suggests that when it comes to cross-lingual transfer, at least in the case of model distillation, lexicalization differences across languages may be more crucial than other linguistic factors such as word order. This finding offers valuable guidance for optimizing cross-lingual transfer systems.

Limitations

Our work has several limitations (we intend to address them in future work). First, all our experiments are conducted using a monolingual teacher model. We consider it important to examine the influence of multilingual pretraining. The potential impact of a representation space that is not tailored to a particular language could be substantial. Secondly, the sum of degradations resulting from the various manipulations we apply does not reach the degradation caused by cross-lingual transfer. This could stem from the fact that translations are not always accurate, but it can also indicate that we are missing an important determinant of cross-lingual transferability. Last, many other manipulations that impact the lexical semantics of the source languages were not considered here. For example, it would be valuable to apply the manipulation to a different subset of the language (e.g., enabling only synonymy but not polysemy, filtering by part-of-speech tags etc.).

References

Kabir Ahuja, Shanu Kumar, Sandipan Dandapat, and Monojit Choudhury. 2022. Multi task learning for zero shot performance prediction of multilingual models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5454–5467, Dublin, Ireland. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019a. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Mikel Artetxe and Holger Schwenk. 2019b. Massively multilingual sentence embeddings for zeroshot cross-lingual transfer and beyond. In *Transactions of the Association for Computational Linguistics*, volume 7, page 597–610. MIT Press.

Ofir Arviv, Dmitry Nikolaev, Taelin Karidi, and Omri Abend. 2021. On the relation between syntactic divergence and zero-shot performance. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.

Ofir Arviv, Dmitry Nikolaev, Taelin Karidi, and Omri Abend. 2023. Improving cross-lingual transfer through subtree-aware word reordering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 718–736, Singapore. Association for Computational Linguistics.

Yuan Chai, Yaobo Liang, and Nan Duan. 2022. Cross-lingual ability of multilingual masked language models: A study of language structure. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, page 4702–4712. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 8440–8451. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations". In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the best of cross-lingual transfer: Evidence from POS tagging with over 100 languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685, Dublin, Ireland. Association for Computational Linguistics.

Ameet Deshpande, Partha Talukdar, and Karthik Narasimhan. 2022. When is BERT multilingual? isolating crucial ingredients for cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3610–3623, Seattle, United States. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pretraining 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, page 4171–4186. Association for Computational Linguistics.

Konstantin Dobler and Gerard de Melo. 2023. Focus: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Blazej Dolicki and Gerasimos Spanakis. 2021. Analysing the impact of linguistic features on cross-lingual transfer. *CoRR*, abs/2105.05975.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*,

pages 4423–4437, Online. Association for Computational Linguistics.

Alexandre François. 2008. Semantic maps and the typology of colexification. *From polysemy to semantic change: Towards a typology of lexical semantic associations*, 106:163.

Thanasis Georgakopoulos, Eitan Grossman, Dmitry Nikolaev, and Stéphane Polis. 2022. Universal and macro-areal patterns in the lexicon: A case-study in the perception-cognition domain. *Linguistic Typology*, 26(2):439–487.

João Graça, Joana Paulo Pardal, Luísa Coheur, and Diamantino Caseiro. 2008. Building a golden collection of parallel multi-language word alignment. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

John Harvill, Roxana Girju, and Mark Hasegawa-Johnson. 2022. Syn2Vec: Synset colexification graphs for lexical semantic similarity. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5270, Seattle, United States. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, page 2101–2112. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020b. Xtreme: a massively multilingual multitask benchmark for evaluating cross-lingual generalization. *arXiv preprint*.

Joshua Conrad Jackson, Joseph Watts, Teague R Henry, Johann-Mattis List, Robert Forkel, Peter J Mucha, Simon J Greenhill, Russell D Gray, and Kristen A Lindquist. 2019. Emotion semantics show both cultural variation and universal structure. *Science*, 366(6472):1517–1522.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1627–1643, Online. Association for Computational Linguistics.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for POS tagging without cross-lingual resources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2832–2838, Copenhagen, Denmark. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Johann-Mattis List, Simon J. Greenhill, Cormac Anderson, Thomas Mayer, Tiago Tresoldi, and Robert Forkel. 2018. Clics2: An improved database of cross-linguistic colexifications assembling lexical data with the help of cross-linguistic data formats. *Linguistic Typology*, 22(2):277–306.

Yihong Liu, Haotian Ye, Leonie Weissweiler, and Hinrich Schütze. 2023a. Crosslingual transfer learning for low-resource languages based on multilingual colexification graphs. *arXiv preprint arXiv:2305.12818*.

Yihong Liu, Haotian Ye, Leonie Weissweiler, Philipp Wicke, Renhao Pei, Robert Zangenfeind, and Hinrich Schütze. 2023b. A crosslingual investigation of conceptualization in 1335 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12969–13000, Toronto, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Minh Van Nguyen, Viet Lai, Amir Pouran Ben Veyseh, 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*.

Vaidehi Patil, Partha Talukdar, and Sunita Sarawagi. 2022. Overlap-based vocabulary generation improves cross-lingual transfer among related languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–233, Dublin, Ireland. Association for Computational Linguistics.

Fred Philippy, Siwen Guo, and Shohreh Haddadan. 2023. Towards a common understanding of contributing factors for cross-lingual transfer in multilingual language models: A review. In *Proceedings*

of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5877–5891, Toronto, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert networks. In *Findings of the Association for Computational Linguistics: EMNLP 2019*, page 3982–3992. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, page 3982–3992. Association for Computational Linguistics.

Liang Tian, Derek Wong, Lidia Chao, Paulo Quaresma, Francisco Oliveira, Shuo Li, Yiming Wang, and Yi Lu. 2014. Um-corpus: a large english-chinese parallel corpus for statistical machine translation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, page 4003–4012. European Language Resources Association.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 833–844, Hong Kong, China. Association for Computational Linguistics.

Zhengxuan Wu, Alex Tamkin, and Isabel Papadimitriou. 2024. Oolong: Investigating what makes transfer learning hard with controlled studies.

Yang Xu, Khang Duong, Barbara C Malt, Serena Jiang, and Mahesh Srinivasan. 2020. Conceptual relations predict colexification across languages. *Cognition*, 201:104280.

A Baseline Experiments & Degradation Analysis

A.1 Training the student model on English

To understand how both the size of the data and the selected model architecture influence the quality of alignment we begin by training the student models on English. We train models of various architectures on subsets of various sizes from *CC-100*. The tokenizer we use is the original tokenizer of the teacher model. Table 4a reports the average cosine similarity (ACS) of all the sentences in the separated test set when they are encoded once using the teacher model and once using the student model. Table 4b reports the accuracy on the XNLI test set in a zero-shot setting (the MLP built upon the teacher model achieved an accuracy of 71.2%).

	50K	**100K**	**200K**	**1M**
3 hidden layers and 4 attention heads	0.658	0.727	0.793	0.866
6 hidden layers and 6 attention heads	**0.684**	**0.754**	**0.827**	0.9
9 hidden layers and 8 attention heads	0.682	0.737	0.818	**0.909**
12 hidden layers and 12 attention heads	0.677	0.74	0.81	0.901

(a) Average cosine similarity (ACS) of all the sentences paired with themselves in a held-out subset of *CC-100*.

	50K	**100K**	**200K**	**1M**
3 hidden layers and 4 attention heads	53.7	57.6	61.3	63.3
6 hidden layers and 6 attention heads	55.7	**59.6**	**62.9**	65.7
9 hidden layers and 8 attention heads	**56.2**	58.1	62.9	**65.8**
12 hidden layers and 12 attention heads	54.3	58.1	61.8	64.9

(b) XNLI test accuracy in a zero-shot setting.

Table 4: Results from the distillation process with English as the target language for various architectures and various dataset sizes.

Several conclusions can be drawn. First, we observe a robust correlation (Pearson correlation of 0.988) between the ACS score and zero-shot performance (the intrinsic and extrinsic performance respectively). This proves that the quality of the alignment with the pretrained representation space can be a useful tool for predicting zero-shot performance. Secondly, the results demonstrate that the size of the corpus has a great effect on the quality of the alignment. With 1M sentences, one can already train a student model that achieves an ACS score of 0.909 out of 1. Lastly, results indicate that the architecture of the student model has a relatively minor impact on performance. However, beyond a certain model size, training results reflect overfitting. As our main concern is low-resource languages, we decide to stick with the architecture that shows optimal performance in limited data scenarios: 6 hidden layers and 6 attention heads.

A.2 Cross-lingual Transfer

We conduct a cross-lingual experiment using the *TED-2020* parallel corpus. Results are presented in

Table 5. We present the outcomes of training the English encoders with both the original teacher's model tokenizer and a newly trained tokenizer (see Appendix F).

	ACS score		XNLI accuracy	
	50K	**100K**	**50K**	**100K**
English - teachers' tokenizer	0.74	0.804	56.6	60.5
English - new *CC-100* tokenizer	0.725	0.786	55.7	59.4
Spanish	0.601	0.657	49.4	54
Greek	0.574	0.632	49.9	53.1
Chinese	0.555	0.6	40.9	46.5
Hebrew	0.545	0.606	X	X

Table 5: Results from the distillation process (average cosine similarity scores and XNLI accuracies) for various languages using the *TED-2020* parallel corpus.

We can see that the tokenizer's substitution results in only a minor performance degradation (0.015 points in ACS score when trained with 50K sentences), while the transition to a different language leads to a substantial decrease (0.124 when trained with 50K Spanish sentences). Unsurprisingly, languages closer to English in terms of phylogenetic distance, produce higher ACS scores and better zero-shot performance.

B Word Reordering Algorithm

We hereby describe the word reordering algorithm developed Arviv et al. (2023), that we apply to permute the words of the source sentences so that it will conform to the syntactic structure of the target language. The algorithm relies on the statistics of the Universal Dependencies (UD) treebank to permute the words of a sentence in one language so that they mimic the syntactic structure of another. The algorithm is built on the assumption that a contiguous subsequence, which constitutes a grammatical unit in the original sentence, should remain a contiguous subsequence after reordering, although the order of words within that subsequence may change. It operates, therefore, on a UD dependency tree, recursively permuting each sub-tree so that it will conform to the order of an equivalent sub-tree in the target language.

Within each sub-tree, the reordering is applied based on the notion of *pairwise ordering distributions*. Given a sentence t in a language L_t and its UD parse tree $T(t)$, which contains the set of dependency labels $\pi = (\pi_1, ..., \pi_n)$, Arviv et al. denote the *pairwise ordering distribution* in language L_t of two UD nodes with dependency labels π_i, π_j, in a sub-tree with the root label π_k by:

$$P_{\pi_k,\pi_i,\pi_j} = p; p \in [0,1] \tag{4}$$

where p stands for the probability of a node with a dependency label π_i to be linearly ordered before a node with a label π_j, in a sub-tree with a root of label π_k, in a language L_t.[10]

Given a sub-tree $T_i \in T(t)$, for each of its node pairs, these probabilities are formulated as a constraint:

$$\pi_k : (\pi_i < \pi_j) = \begin{cases} \mathbf{1} & \text{if } P_{\pi_k,\pi_i,\pi_j} \geq 0.5 \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{5}$$

where $\pi_k : (\pi_i < \pi_j) = \mathbf{1}$ indicates that a node with label π_i should be linearly ordered before a node with label π_j if they are direct children of a node with label π_k. A constraint is said to be satisfied if and only if the node with label π_i is indeed positioned in the sentence before the node with label π_j. For each individual sub-tree T_i, all its pairwise constrains are extracted, and an SMT solver is used to compute a legal ordering which satisfies all the constraints.[11]

C Lexical manipulation: Implementation Details

Tokenization and Lemmatization. Before we perform word-to-word alignment, we have to separate the sentences' tokens and lemmatize them. For this purpose we use *Trankit* (Nguyen et al., 2021), a multilingual NLP toolkit based on *XLM-R*. For Simplified Chinese, however, we prefer the Jieba tokenizer.

Graph Filtering. When considering the filtering of the graph, we face two choices: we can either apply identical parameters for all languages or customize parameters for each language in a way that ensures a similar percentage of alignment instances is filtered from the graph. The first option maintains a similar level of noise across languages but has a drawback: when we apply the lexical manipulation, removing a high percentage of alignment instances from the graph results in selecting the

[10] Note that a single node can act both as a representative of its sub-tree and the head of that sub-tree.

[11] If it is not possible to fulfill all the constraints, the algorithm maintains the original order of the sub-tree.

most common word too frequently (see step 4c in the lexical manipulation procedure), and therefore loses the ability to make meaningful comparisons across different languages.

In our chosen method, we aim for the middle ground. We start by removing from the graph every edge with a weight below the threshold of 5 to exclude matches that are not substantial. Then, for each language, we set a specific threshold to remove edges whose weight is relatively small compared to other edges originating from the same vertex. We set this second threshold in such a way that for each language, a total of approximately 12% of the alignment instances are filtered out. In the case of Spanish and Greek, the appropriate threshold is 2%, while for Simplified Chinese, it is 0.15%.

D Comparing Alignments to Gold Standard

We evaluate the alignment results of our algorithm against the original Simalign alignments, using the gold standard provided by (Graça et al., 2008). We focus our comparison on the *English–Spanish* alignments, as this language pair is the sole one utilized in our research. The obtained results are as presented in Table 6. We can see that our algorithm's outputs achieve higher precision but lower recall, resulting in a slightly better F1 score overall.

To further understand this point, let us examine a specific example (as others are similar): the sentence *We take note of your statement* is translated into *Tomamos nota de esa declaración*. While the Simalign auto-aligner aligns *your* with *esa*, our algorithm filters out this alignment, as these two words are rarely translations of one another in the larger corpus. Although we miss a correct alignment in the gold standard, this approach conforms to our goal of mapping lexicons consistently.

	Precision	**Recall**	**F1**
Original simalign	73.39	**90.8**	81.17
Our algorithm	**76.55**	86.58	**81.26**

Table 6: Comparison of Alignments to Gold Standard

E Datasets

TED. The bitext we use for training and intrinsic evaluation of the student models throughout all manipulation experiments is the *TED-2020* parallel corpus (Reimers and Gurevych, 2020). This corpus contains a crawl of nearly 4000 TED transcripts from July 2020, which have been translated into over 100 languages by a global community of volunteers. We select this corpus because it contains languages from different language families, and because its translations are of relatively high quality. To further simplify it, we lowercase the entire dataset and filter it to include only sentences with familiar characters, up to one punctuation mark, and word counts ranging from 4 to 16.[12] The corpus is licensed under CC BY–NC–ND 4.0.

CC-100. When we require a larger corpus, but not necessarily a parallel one, we turn to the *CC-100* corpus (Wenzek et al., 2020). This corpus serves us for training the tokenizers (See Appendix F). Our goal in tokenizer training is to prevent overspecialization on the limited number of sentences used for training the student model. To achieve this, we use the largest and most diverse corpus available to us, namely, CC-100. Our experiments simulate scenarios where there is a significant amount of monolingual data for a specific language but minimal parallel data, which is the case for many low-resource languages. Additionally, we employ this corpus for our *English–English* experiments (see Appendix A.1). In both cases, we apply the same simplifying process as for TED, bringing the formats of the two datasets closer to each other. The license of this corpus is unspecified by the authors, but they state that they will make it publicly available.

XNLI. For extrinsic evaluation we use Natural Language Inference (NLI), as it is a well-known sentence level semantic task. The task is to determine the inference relation between two sentences: *entailment*, *contradiction*, or *neutral*. The corpus we use is the *Cross-lingual Natural Language Inference (XNLI)* (Conneau et al., 2018), which contains 15 different languages. There is no need to apply a simplifying process to this dataset, as the sentences are already relatively short and do not contain unconventional characters. The corpus is licensed under CC BY-NC 4.0.

Europarl. In order to extract a bipartite graph which is statistically meaningful for our lexical manipulation, we require a large parallel corpus. We use *Europarl*, which consists of the proceedings of the European Parliament from 1996 to 2012. This corpus contains only European languages, so

[12] Except for Simplified Chinese, where, due to the different nature of logographic writing systems, we filter by counting 5-25 symbols.

we must turn to other sources when experimenting with languages from different language groups. The corpus is freely available.

UM. We extract our *English–Chinese* bipartite graph from the *UM* parallel corpus (Tian et al., 2014). It contains more than 2M *English–Chinese* sentence pairs from a great variety of domains. The corpus is licensed under CC BY–NC–ND 4.0.

F Tokenizers

When training English models, we examine two different tokenizers: the original teachers' tokenizer, and a new tokenizer we train on the simplified *CC-100* corpus. In the case of other languages, we train a new tokenizer on the simplified *CC-100* corpus.

The cases of the script and lexical manipulations each require its special treatment. In the case of the script manipulations, we create an artificial language which is composed of English words with foreign symbols, so we require a tokenizer which is familiar with this specific language. We simply apply the manipulation to the English *CC-100* corpus and train a tokenizer on the transformed sentences.

In the case of the lexical manipulation, we swap some English words while retaining others, resulting in an artificial language which is a fusion of two languages. Therefore, a bilingual tokenizer is required. We train a bilingual tokenizer for each language pair using the *CC-100* corpus.[13]

[13]Note that the word-reorder manipulation, as it maintains the same set of words as in the original sentence, does not require any special treatment.