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 modification 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 the alignment quality, the degree of lexical matching between the two languages, which we define using a measure of translation entropy, greatly affects it.

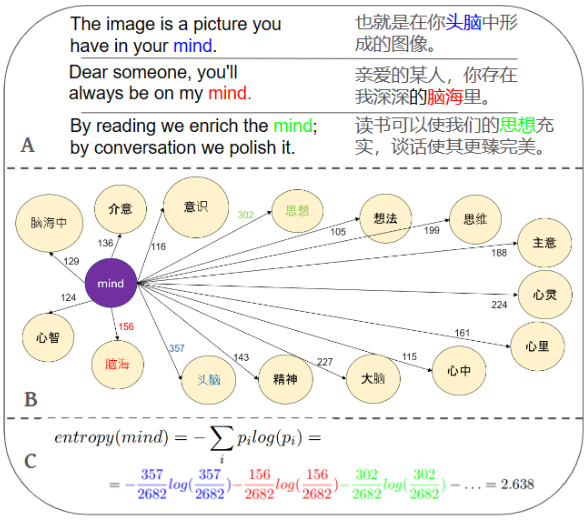

The image is a picture you have in your mind.
也就是在你头脑中形成的图像。
Dear someone, you'll always be on my mind.
亲爱的某人，你存在我深深的脑海里。
By reading we enrich the mind; by conversation we polish it.
读书可以使我们的思想充实，谈话使其更臻完美。
A
脑海中
介意
意识
思想
想法
思维
主意
mind
心灵
心智
脑海
头脑
精神
大脑
心中
心里
302
105
199
188
136
116
129
124
156
357
143
227
115
161
224
B
$entropy(mind) = -\sum_i p_i log(p_i) =$
$= -\frac{357}{2682}log(\frac{357}{2682}) - \frac{156}{2682}log(\frac{156}{2682}) - \frac{302}{2682}log(\frac{302}{2682}) - \ldots = 2.638$
C

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 distribute meanings across their vocabularies in unique ways. In English, the concept *wall* encompasses both a structural component in a house and a defensive barrier around a city, whereas Spanish distinguishes between 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.

While multilingual NLP is gaining increasing attention, a performance gap persists between English and other languages, particularly low-resource ones. To address this, various cross-lingual transfer techniques have been proposed (Kim et al., 2017; Artetxe and Schwenk, 2019a; Dobler and de Melo, 2023), including training shared multilingual representation spaces (Artetxe et al., 2018; Ruder et al., 2019; Heffernan et al., 2022; Tan et al., 2023).

In this work, we aim to understand 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 from one another and evaluate their individual impact, we perform manipulations to the English language in order to 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 on 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 also 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 Masked Language Models (MMLMs) 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) introduced a novel architecture. They initially trained a base model (SentenceBERT) using a sentence-level objective to obtain sentence representations (2019). Subsequently, they 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

Several studies have been conducted to explore the factors that impact the effectiveness of zero-shot cross-lingual transfer. Cotterell and Heigold (2017) showed that the zero-shot performs better when applied to languages within the same language family. Arviv et al. (2021) demonstrated a correlation between the preservation of syntactic relations in translation and zero-shot performance.

Chai et al. (2022) conducted a similar experiment but under controlled laboratory conditions. In order to evaluate the impact of specific language properties, they create an artificial language by modifying English and pretrain a bilingual MLM with English and its modified version. Their focus was on syntactic properties, particularly word order. They observed a negligible impact on zero-shot performance when examining subtle modifications in constituent order, but a significant impact when shuffling the entire sentence randomly.

While previous experiments focus on syntactic features, they do not address the semantic aspect. In this paper, we focus on lexical semantics.

2.3 Lexicalisation Patterns

Lexicalisation 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 follow Chai et al. (2022) and 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 relatively 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:

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

where λ is a margin parameter we always set to 0. 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 similarity score 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 comprehensive approach, investigating how lexical semantics fit into the larger context of language properties. We broadly categorize linguistic properties into three distinct aspects: script, syntax, and lexical semantics. For each of these aspects, we define a manipulation that solely modifies it. In the first case, we substitute the letters of the English alphabet with symbols of a different script to assess the impact of the script (Section §4.1). In the second case, we rearrange the word order in sentences, thus examining the effect of the syntactic structure, or at least a specific aspect of it (Section §4.2). Finally, we replace the English lexicon with that of a target language to explore the significance of variations in lexicalization patterns (Section §4.3). By isolating each linguistic aspect, we intend to get a clear 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: $a \rightarrow \alpha$,

$b \to \beta, c \to \gamma$, and so forth. This way, the sentence *Brown cows eat grass* will transform into: $\beta\sigma o\psi\xi$ $\gamma o\psi\tau$ $\epsilon\alpha\upsilon$ $\eta\sigma\alpha\tau\tau$.

4.2 Manipulating Word Order

The second manipulation we use is a word reordering one. We apply the word reordering algorithm developed 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* (see its dependency tree in Figure 3) 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, outputs: *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 an 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, attaining word-aligned bitext poses a significant challenge. While manually aligned parallel datasets are scarce and limited in size, model-based automatic aligners are prone to noise, often aligning unrelated words. In the process of defining a lexical manipulation that relies on mapping one lexicon to another, it is crucial to ensure consistent mappings while reducing noisy matches, thus giving priority to precision over recall. To achieve this, we follow a careful process that involves extracting a bipartite graph from a bitext.

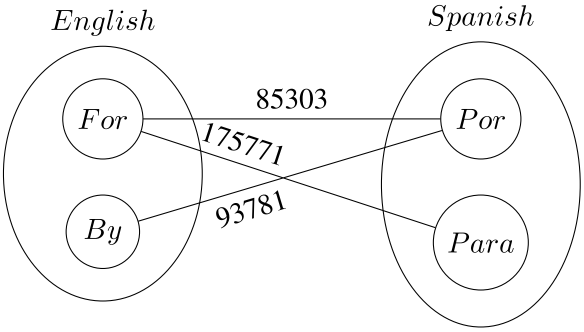


Figure 2: Illustration of the weighted sub-graph which results from the *Europarl* parallel corpus. The edges represent the possibility that two words are translations of each other. The weights denote the number of occurrences that each word pair is aligned in the bitext.

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 therefore 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 align in a one-to-one manner, 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 (see de-

[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] It is worth mentioning that 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.

tails in Appendix C) to a large $L_s - L_t$ parallel corpus, extracting a weighted bipartite graph G as described above.

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 incorrect output of the automatic aligner shown in Figure 3. When applying our swapping algorithm, we first check whether the edges $(brown - el)$ and $(eat - comer)$[6] appear in the bipartite graph. As only the second edge is present, we replace the word *eat* with $comer$. Next, we search for words in the target sentence that are linked to the source words in the graph, resulting in the edges $(brown - marr\acute{o}n)$, $(cow - vaca)$, and $(grass - hierba)$. These three words are swapped with their corresponding pairs as well. This process ultimately yields the sentence: *Marrones vacas comen hierba.*

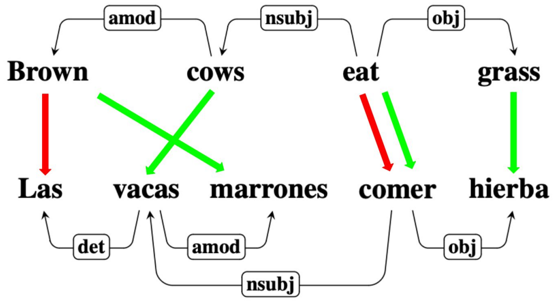


Figure 3: Comparison between an incorrect output of the automatic aligner (red) and the corrected alignment we employ (green), presented with UD tree annotation for each sentence.

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 4A) and another for the target words (Figure 4B). In the first, we compute the *translation entropy* for all source vertices $v \in V_s$ and partition

[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] We compare the precision and recall of our alignments with those of the automatic aligner against the gold standard. We observe higher precision but lower recall, resulting in a slightly better F1 score overall in our alignments. For further details please refer to Appendix D.

[6] The lemmas of las and $comer$ respectively.

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

the set V_s into three disjoint subsets based on the percentile of the *translation entropy* values. 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]

Returning to the example from Figure 3, 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 we use in our work. For further details regarding the datasets and the rationale behind their selection and usage, please refer to Appendix E. The primary bitext we use for training is the *TED-2020* parallel corpus (Reimers and Gurevych, 2020). When we require a larger corpus, but not necessarily a parallel one, we turn to the *CC-100* corpus (Wenzek et al., 2020). 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 parallel corpus, and therefore turn to the *Europarl* parallel corpus. As this corpus contains only European languages, we extract the $English - Chinese$ bipartite graph from the *UM* parallel corpus (Tian et al., 2014).

5.2 Models

Teacher Model. We select the pretrained sentence transformer *a*ll-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. In all cases, 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.[9]

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.

6 Experiments & Results

To assess the individual impact of each linguistic property on cross-lingual transfer, we apply our manipulations to English and carry out the distillation process for each artificial language L_A. For intrinsic evaluation we rely on the similarity score defined in Section §3, and for extrinsic evaluation, we use XNLI zero-shot accuracy.

Before manipulating English, we conduct experiments to obtain reference points for evaluating the models. First, we perform the distillation process on regular English sentences. 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 average

[8] It is worth mentioning that the two configurations are not directly comparable: removing 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] For details regarding the tokenizers we employ, see Appendix F.

similarity score, 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 average similarity score, 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 average similarity score, 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, replacing the English characters with Greek and Simplified Chinese characters, sorted by their frequency.The results, reported in Table 1, show no degradation in performance when compared to the encoders trained with a new tokenizer (see Appendix F). This suggests that substituting the script has a similar effect as replacing the tokenizer.

	Similarity score		XNLI accuracy	
	50K	**100K**	**50K**	**100K**
English (new tok.)	0.725	0.786	55.7	59.4
Greek alphabet	0.725	0.786	54.6	58.7
Chinese symbols	0.728	0.788	55.3	58.4

Table 1: Results from the distillation process for the script substitution experiment.

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 average similarity 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%.

	Similarity score		XNLI accuracy	
	50K	**100K**	**50K**	**100K**
English (new tok.)	0.725	0.786	55.7	59.4
Spanish order	0.722	0.779	55.9	58.9
Greek order	0.718	0.773	54.8	58.1
Chinese order	0.723	0.774	55.1	58.1
Hebrew order	0.725	0.781	55.2	59
Hindi order	0.72	0.776	56.5	57.9

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

6.3 Lexical Swapping

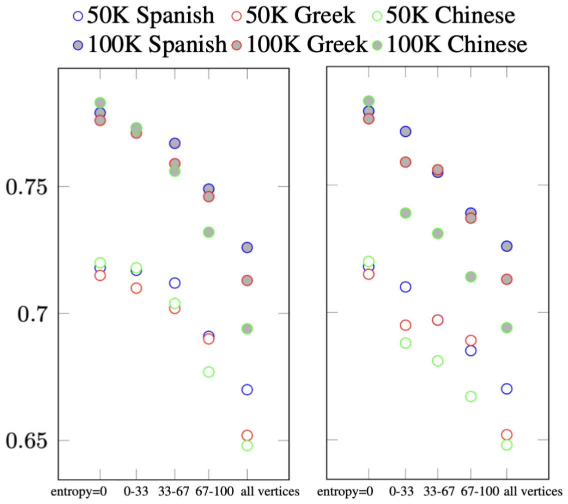
We follow the steps described in Section §4.3 for Spanish, Greek and Simplified Chinese. When constructing the weighted bipartite graph, for Spanish and Greek we use the datasets *Europarl+TED*, whereas for Simplified Chinese we use *UM+TED*. Results are presented in Table 3. In this experiment, we observe a significant decrease in both the average similarity score and the XNLI accuracy. The language that performs the worst is Simplified Chinese, with up to 0.091 degradation in the average similarity score and up to 5.9% in XNLI accuracy.

	Similarity score		XNLI accuracy	
	50K	**100K**	**50K**	**100K**
English (new tok.)	0.725	0.786	55.7	59.4
Spanish lexicon	0.67	0.726	53.4	57.5
Greek lexicon	0.652	0.713	51.6	56.1
Chinese lexicon	0.646	0.694	50.9	53.5

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, and as a result they affect performance in cross-lingual transfer tasks. To gain a deeper understanding of this phenomenon, we proceed to apply the same manipulation, this time selectively swapping only a subset of the words in 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 4A presents the outcome of filtering the source vertices, and Figure 4B 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.



(A) Source (B) Target

Figure 4: 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 scale represents the entropy values ranging from 0 to *all vertices*, with values between 0 and *all* indicating percentiles dividing the set of vertices.

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 words with low entropy, it has minimal impact on performance, but when we swap 33% of words 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 altered by different words in different contexts, whereas in the case of polysemy, several distinct words are altered by the same word. The first experiment (filtering the source words) mostly simulates the impact of the synonymy phenomenon, while the second experiment (filtering the target words) mostly simulates the impact of the 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.

We first apply a script substitution and observe no degradation in performance. We then examine the impact of word order. Unlike Chai et al. (2022), who made only subtle modifications to the constituent order in some experiments and shuffled all the words in the sentence with others, we apply manipulations that permute many words in the sentence while still maintaining a coherent syntactic structure. We believe that while Chai et al.'s conclusion is overly broad in its scope, this manipulation provides us with a more nuanced understanding of the role of word order in cross-lingual transfer. Our initial observations imply that word order differences, if systematic, may not play a crucial role.

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 relative to words with lower entropy. These findings validate our hypothesis: the more the lexicons align in a one-to-one manner, the better cross-lingual transfer will perform.

To conclude, among the three manipulations we apply, the only one that was found to have a substantial impact is the lexical swapping manipulation. This suggests that when it comes to cross-lingual transfer, at least in the case of model distillation, the difference between the lexical semantics of the languages may be more crucial than other linguistic factors such as word order. This insight highlights the importance of lexical compatibility and 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 a piece of the puzzle. Lastly, we think it will be valuable to further analyze the lexical manipulation, maybe by applying it to a different subset of the language (enabling only synonymy but not polysemy, filtering by part-of-speech, etc.).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. 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.

Avihay Chriqui and Inbal Yahav. 2022. Hebert & hebemo: a hebrew bert model and a tool for polarity analysis and emotion recognition. *INFORMS Journal on Data Science*.

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.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2017*, page 748–759. 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.

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.

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 light-weight 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*.

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.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Weiting Tan, Kevin Heffernan, Holger Schwenk, and Philipp Koehn. 2023. Multilingual representation distillation with contrastive learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1477–1490, Dubrovnik, Croatia. 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.

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 similarity score 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 similarity score of all the sentences paired with themselves in a separate 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 average similarity scores 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 average similarity 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, it is worth noting that beyond a certain model size, training results are 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 also present the outcomes of training the English encoders with a newly trained tokenizer (see Appendix F).

	Similarity 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 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 similarity 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 similarity 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.

Automatic Aligner. To obtain 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 from one sentence to those of another. We run it with *XLM-R* as the base model, and set the matching method to be $ArgMax$.

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 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 alignments 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

[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.

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 primary bitext we use for training 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 have selected this corpus because it contains languages from different language families, and because its translations are of relatively high quality. To further simplify it, we convert the entire dataset to lowercase and filter it to include only sentences with familiar characters, up to one punctuation mark, and word counts ranging from 4 to 16.[12]

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 our $English - English$ experiments (see Section §A.1). We apply the same simplifying process as for TED, bringing the formats of the two datasets closer to each other.

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.

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 we must turn to other sources when experimenting with languages from different language groups.

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.

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 language, we train a new tokenizer on the simplified *CC-100* corpus, except for Hebrew, where we use the tokenizer from the *HeBERT* pretrained model (Chriqui and Yahav, 2022), and Chinese, where we use the tokenizer from the *Bert-Base-Chinese* pretrained model.

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 transformer for each language pair using the *CC-100* corpus, except for $English - Chinese$, for which we use the *UM* corpus.[13]

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

[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.