Everything Is All It Takes: A Multipronged Strategy for Zero-Shot Cross-Lingual Information Extraction

Mahsa Yarmohammadi[1]*, Shijie Wu[1]*, Marc Marone[1], Haoran Xu[1], Seth Ebner[1], Guanghui Qin[1], Yunmo Chen[1], Jialiang Guo[1], Craig Harman[1], Kenton Murray[1], Aaron Steven White[2], Mark Dredze[1], Benjamin Van Durme[1]

[1]Johns Hopkins University, [2]University of Rochester

{mahsa,shijie.wu,vandurme}@jhu.edu

Abstract

Zero-shot cross-lingual information extraction (IE) describes the construction of an IE model for some target language, given existing annotations exclusively in some other language, typically English. While the advance of pretrained multilingual encoders suggests an easy optimism of "train on English, run on any language", we find through a thorough exploration and extension of techniques that a combination of approaches, both new and old, leads to better performance than any one cross-lingual strategy in particular. We explore techniques including data projection and self-training, and how different pretrained encoders impact them. We use English-to-Arabic IE as our initial example, demonstrating strong performance in this setting for event extraction, named entity recognition, part-of-speech tagging, and dependency parsing. We then apply data projection and self-training to three tasks across eight target languages. Because no single set of techniques performs the best across all tasks, we encourage practitioners to explore various configurations of the techniques described in this work when seeking to improve on zero-shot training.

1 Introduction

We consider zero-shot cross-lingual information extraction (IE), in which training data exists in a source language but not in a target language. Massively multilingual encoders like Multilingual BERT (mBERT; Devlin et al., 2019) and XLM-RoBERTa (XLM-R; Conneau et al., 2020a) allow for a strategy of training only on the source language data, trusting entirely in a shared underlying feature representation across languages (Wu and Dredze, 2019; Conneau et al., 2020b). However, in meta-benchmarks like XTREME (Hu et al., 2020), such cross-lingual performance on structured prediction tasks is far behind that on sentence-level or retrieval tasks (Ruder et al., 2021): performance in the target language is often far below that of the source language. Before multilingual encoders, cross-lingual IE was approached largely as a data projection problem: one either translated and aligned the source training data to the target language, or at test time one translated target language inputs to the source language for prediction (Yarowsky and Ngai, 2001).

We show that by augmenting the source language training data with data in the target language—either via projection of the source data to the target language (so-called "silver" data) or via self-training with translated text—zero-shot performance can be improved. Further improvements might come from using better pretrained encoders or improving on a projection strategy through better automatic translation models or better alignment models. In this paper, we explore all the options above, finding that *everything is all it takes* to achieve our best experimental results, suggesting that a silver bullet strategy does not currently exist.

Specifically, we evaluate: cross-lingual data projection techniques with different machine translation and word alignment components, the impact of bilingual and multilingual contextualized encoders on each data projection component, and the use of different encoders in task-specific models. We also offer suggestions for practitioners operating under different computation budgets on four tasks: event extraction, named entity recognition, part-of-speech tagging, and dependency parsing, following recent work that uses English-to-Arabic tasks as a test bed (Lan et al., 2020). We then apply data projection and self-training to three structured prediction tasks—named entity recognition, part-of-speech tagging, and dependency parsing—in multiple target languages. Additionally, we use self-training as a control against data projection to determine in which situations data projection improves performance.

*Equal contribution

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1950–1967
November 7–11, 2021. ©2021 Association for Computational Linguistics

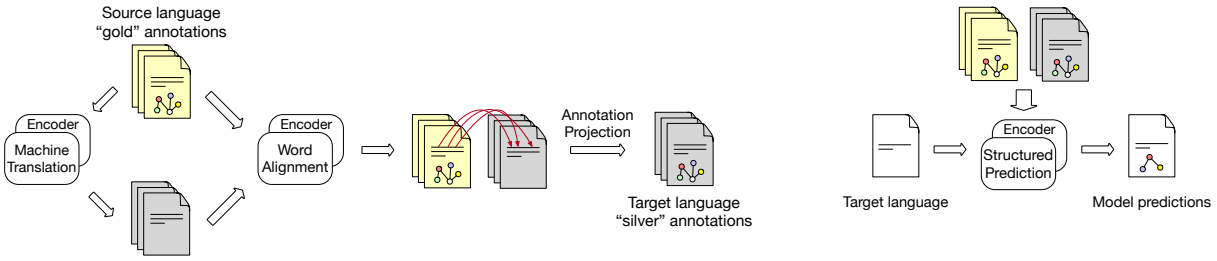

Figure 1: Process for creating projected “silver” data from source “gold” data (left). Downstream models are trained on a combination of gold and silver data (right). Components in boxes have learned parameters.

Our contributions include the following:

- examination of the impact of statistical and neural word aligners and publicly available and custom machine translation (MT) models on annotation projection,
- examination of the impact of publicly available and custom multilingual and bilingual encoders of different model sizes, both as the basis of models for downstream tasks and as components of word aligners and MT models,
- use of self-training on translated text as a way to automatically create labeled target language data and as a controlled comparison to analyze when data projection helps, and
- extensive experiments demonstrating improvements over zero-shot transfer and analysis showing that the best setup is task dependent.

We also make available models and tools that enabled our analysis.

2 Universal Encoders

While massively multilingual encoders like mBERT and XLM-R enable strong zero-shot cross-lingual performance (Wu and Dredze, 2019; Conneau et al., 2020a), they suffer from the curse of multilinguality (Conneau et al., 2020a): cross-lingual effectiveness suffers as the number of supported languages increases for a fixed model size. We would therefore expect that when restricted to only the source and target languages, a bilingual model should perform better than (or at least on par with) a multilingual model of the same size, assuming both languages have sufficient corpora (Wu and Dredze, 2020a). If a practitioner is interested in only a small subset of the supported languages, *is the multilingual model still the best option*?

To answer this question, we use English and Arabic as a test bed. In Table 1, we summarize existing publicly available encoders that support both English and Arabic.[1] Base models are 12-layer Transformers (`d_model` = 768), and large models are 24-layer Transformers (`d_model` = 1024) (Vaswani et al., 2017). As there is no publicly available large English–Arabic bilingual encoder, we train two encoders from scratch, named L64K and L128K, with vocabulary sizes of 64K and 128K, respectively.[2] With these encoders, we can determine the impacts of model size and the number of supported languages.

3 Data Projection

We create silver versions of the data by automatically projecting annotations from source English gold data to their corresponding machine translations in the target language.[3] Data projection transfers word-level annotations in a source language to a target language via word-to-word alignments (Yarowsky et al., 2001). The technique has been used to create cross-lingual datasets for a variety of structured natural language processing tasks, including named entity recognition (Stengel-Eskin et al., 2019) and semantic role labeling (Akbik et al., 2015; Aminian et al., 2017; Fei et al., 2020).

To create silver data, as shown in Figure 1, we: (1) translate the source text to the target language using the MT system described in Section 5.2, (2) obtain word alignments between the original and translated parallel text using a word alignment tool, and (3) project the annotations along the word alignments. We then combine silver target data with gold source data to augment the training set for the structured prediction task.

For step (1), we rely on a variety of source-to-

[1] We do not include multilingual T5 (Xue et al., 2021) as it is still an open question on how to best utilize text-to-text models for structured prediction tasks (Ruder et al., 2021).

[2] L128K available at https://huggingface.co/jhu-clsp/roberta-large-eng-ara-128k

[3] Code available at https://github.com/shijie-wu/crosslingual-nlp

	Base	Large
Multilingual	mBERT (Devlin et al.)	XLM-R (Conneau et al.)
Bilingual	GBv4 (Lan et al.)	L64K & L128K **(Ours)**

Table 1: Encoders supporting English and Arabic.

target MT systems. To potentially leverage monolingual data, as well as contextualized cross-lingual information from pretrained encoders, we feed the outputs of the final layer of frozen pretrained encoders as the inputs to the MT encoders. We consider machine translation systems: (i) whose parameters are randomly initialized, (ii) that incorporate information from massively multilingual encoders, and (iii) that incorporate information from bilingual encoders that have been trained on only the source and target languages.

After translating source sentences to the target language, in step (2) we obtain a mapping of the source words to the target words using publicly available automatic word alignment tools. Similarly to our MT systems, we incorporate contextual encoders in the word aligner. We hypothesize that better word alignment yields better silver data, and better information extraction consequently.

For step (3), we apply direct projection to transfer labels from source sentences to target sentences according to the word alignments. Each target token receives the label of the source token aligned to it (token-based projection). For multi-token spans, the target span is a contiguous span containing all aligned tokens from the same source span (span-based projection), potentially including tokens not aligned to the source span in the middle. Three of the IE tasks we consider—ACE, named entity recognition, and BETTER—use span-based projection, and we filter out projected target spans that are five times longer than the source spans. Two syntactic tasks—POS tagging and dependency parsing—use token-based projection. For dependency parsing, following Tiedemann et al. (2014), we adapt the disambiguation of many-to-one mappings by choosing as the head the node that is highest up in the dependency tree. In the case of a non-aligned dependency head, we choose the closest aligned ancestor as the head.

To address issues like translation shift, filtered projection (Akbik et al., 2015; Aminian et al., 2017) has been proposed to obtain higher precision but lower recall projected data. To maintain the same amount of silver data as gold data, in this study we do not use any task-specific filtered projection methods to remove any sentence.

4 Tasks

We employ our silver dataset creation approach on a variety of tasks.[4] For English–Arabic experiments, we consider ACE, BETTER, NER, POS tagging, and dependency parsing. For multilingual experiments, we consider NER, POS tagging, and dependency parsing. We use English as the source language and 8 typologically diverse target languages: Arabic, German, Spanish, French, Hindi, Russian, Vietnamese, and Chinese. Because of the high variance of cross-lingual transfer (Wu and Dredze, 2020b), we report the average test performance of three runs with different predefined random seeds (except for ACE).[5] For model selection and development, we use the English dev set in the zero-shot scenario and the combined English dev and silver dev sets in the silver data scenario.

4.1 ACE

Automatic Content Extraction (ACE) 2005 (Walker et al., 2006) provides named entity, relation, and event annotations for English, Chinese, and Arabic. We conduct experiments on English as the source language and Arabic as the target language. We use the OneIE framework (Lin et al., 2020), a joint neural model for information extraction, which has shown state-of-the-art results on all subtasks. We use the same hyperparameters as in Lin et al. (2020) for all of our experiments. We use the OneIE scoring tool to evaluate the prediction of entities, relations, event triggers, event arguments, and argument roles. For English, we use the same English document splits as (Lin et al., 2020). That work does not consider Arabic, so for Arabic we use the document splits from (Lan et al., 2020).

4.2 Named Entity Recognition

We use WikiAnn (Pan et al., 2017) for English–Arabic and multilingual experiments. The labeling scheme is BIO with 3 types of named entities: PER, LOC, and ORG. On top of the encoder, we use a linear classification layer with softmax to obtain word-level predictions. The labeling is word-level while the encoders operate at subword-level, thus, we mask the prediction of all subwords except for

[4] See Appendix A for dataset statistics and fine-tuning hyperparameters for each task.

[5] We report one run for ACE due to long fine-tuning time.

the first one. We evaluate NER performance by F1 score of the predicted entity.

4.3 Part-of-speech Tagging

We use the Universal Dependencies (UD) Treebank (v2.7; Zeman et al., 2020).[6] Similar to NER, we use a word-level linear classifier on top of the encoder, and evaluate performance by the accuracy of predicted POS tags.

4.4 Dependency Parsing

We use the same treebanks as the POS tagging task. For the task-specific layer, we use the graph-based parser of Dozat and Manning (2016), but replace their LSTM encoder with our encoders of interest. We follow the same policy as that in NER for masking non-first subwords. We predict only the universal dependency labels, and we evaluate performance by labeled attachment score (LAS).

4.5 BETTER

The Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program[7] develops methods for extracting increasingly fine-grained semantic information in a target language, given gold annotations only in English. We focus on the coarsest "Abstract" level, where the goal is to identify events and their agents and patients. The documents come from the news-specific portion of Common Crawl. We report the program-defined "combined F1" metric, which is the product of "event match F1" and "argument match F1", which are based on an alignment of predicted and reference event structures.

To find all events in a sentence and their corresponding arguments, we model the structure of the events as a tree, where event triggers are children of the "virtual root" of the sentence and arguments are children of event triggers (Cai et al., 2018). Each node is associated with a span in the text and is labeled with an event or argument type label.

We use a model for event structure prediction that has three major components: a contextualized encoder, tagger, and typer (Xia et al., 2021).[8] The tagger is a BiLSTM-CRF BIO tagger (Panchendrarajan and Amaresan, 2018) trained to predict child spans conditioned on parent spans and labels. The typer is a feedforward network whose inputs are a parent span representation, parent label embedding, and child span representation. The tree is produced level-wise at inference time, first predicting event triggers, typing them, and then predicting arguments conditioned on the typed triggers.

5 Experiments

5.1 Universal Encoders

We train two English–Arabic bilingual encoders.[9] Both of them are 24-layer Transformers (`d_model` = 1024), the same size as XLM-R large. We use the same Common Crawl corpus as XLM-R for pretraining. Additionally, we also use English and Arabic Wikipedia, Arabic Gigaword (Parker et al., 2011), Arabic OSCAR (Ortiz Suárez et al., 2020), Arabic News Corpus (El-Khair, 2016), and Arabic OSIAN (Zeroual et al., 2019). In total, we train with 9.2B words of Arabic text and 26.8B words of English text, more than either XLM-R (2.9B words/23.6B words) or GBv4 (4.3B words/6.1B words).[10] We build two English–Arabic joint vocabularies using SentencePiece (Kudo and Richardson, 2018), resulting in two encoders: **L64K** and **L128K**. For the latter, we additionally enforce coverage of all Arabic characters after normalization.

5.2 Machine Translation

For all of our MT experiments, we use a dataset of 2M sentences from publicly available data including the UN corpus, Global Voices, wikimatrix, and newscommentary11 (Ziemski et al., 2016; Prokopidis et al., 2016; Schwenk et al., 2021; Callison-Burch et al., 2011). We pre-filtered the data using LASER scores to ensure high quality translations are used for our bitext (Schwenk and Douze, 2017; Thompson and Post, 2020).

All of our systems are based on the Transformer architecture (Vaswani et al., 2017).[11] Our baseline system uses a joint English–Arabic vocabulary with 32k BPE operations (Sennrich et al., 2016). The public system is a publicly released model that has been demonstrated to perform well (Tiedemann, 2020).[12] The other systems use contextualized embeddings from frozen pretrained language models

[6] We use the following treebanks: Arabic-PADT, German-GSD, English-EWT, Spanish-GSD, French-GSD, Hindi-HDTB, Russian-GSD, Vietnamese-VTB, and Chinese-GSD.

[7] https://www.iarpa.gov/index.php/research-programs/better

[8] Code available at https://github.com/hiaoxui/span-finder

[9] Details of pretraining can be found in Appendix B.

[10] We measure word count with `wc -w`.

[11] See Appendix C for a full list of hyperparameters.

[12] The public MT model is available at https://huggingface.co/Helsinki-NLP/opus-mt-en-ar

as inputs to the encoder. For the decoder vocabulary, these systems all use the GBv4 vocabulary regardless of which pretrained language model was used to augment the encoder.

Incorporating Pretrained LMs In order to make use of the pretrained language models, we use the output of the last layer of the encoder. A traditional NMT system uses a prespecified, fixed size vocabulary with randomly initialized parameters for the source embedding layer. To incorporate a pretrained language model, we instead use the exact vocabulary of that model. A sentence is fed into the encoder and the resultant vectors from the output layer are used instead of the randomly initialized embedding layer. We freeze these pretrained language models so that no gradient updates are applied to them during MT training, whereas the randomly initialized baselines are updated. A preliminary experiment in Zhu et al. (2020) uses a related system that leverages the last layer of BERT. However, that experiment was monolingual, and our hypothesis is that the shared embedding space of a multilingual encoder will aid in training a translation system.

Denormalization System Generating text in Arabic is a notoriously difficult problem due to data sparsity problems arising from the morphological richness of the language, frequently necessitating destructive normalization schemes during training that must be heuristically undone in post-processing to ensure well-formed text (Sajjad et al., 2013). All of the most common multilingual pretrained encoders use a form of destructive normalization which removes diacritics, which causes MT systems to translate into normalized Arabic text. To generate valid Arabic text, we train a sequence-to-sequence model that transduces normalized text into unnormalized text using the Arabic side of our bitext, before and after normalization. Our transducer uses the same architecture and hyperparameters as our baseline MT system, but with 1k BPE operations instead of 32k. On an internal held-out test set, we get a BLEU score of 96.9 with a unigram score of 98.6, implying few errors will propagate due to the denormalization process.[13]

Intrinsic Evaluation Table 2 shows the denormalized and detokenized BLEU scores for English–Arabic MT systems with different encoders on the IWLST'17 test set using sacreBLEU (Post, 2018). The use of contextualized embeddings from pretrained encoders results in better performance than using a standard randomly initialized MT model regardless of which encoder is used. The best performing system uses our bilingual L64K encoder, but all pretrained encoder-based systems perform well and within 0.5 BLEU points of each other. We hypothesize that the MT systems are able to leverage the shared embedding spaces of the pretrained language models in order to assist with translation.

Encoder	BLEU
Public	12.7
None	14.9
mBERT	15.7
GBv4	15.7
XLM-R	16.0
L64K	**16.2**
L128K	15.8

Table 2: BLEU scores of MT systems with different pre-trained encoders on English–Arabic IWSLT'17.

5.3 Word Alignment

Until recently, alignments have typically been obtained using unsupervised statistical models such as GIZA++ (Och and Ney, 2003) and fast-align (Dyer et al., 2013). Recent work has focused on using the similarities between contextualized embeddings to obtain alignments (Jalili Sabet et al., 2020; Daza and Frank, 2020; Dou and Neubig, 2021), achieving state-of-the-art performance.

We use two automatic word alignment tools: fast-align, a widely used statistical alignment tool based on IBM models (Brown et al., 1993); and Awesome-align (Dou and Neubig, 2021), a contextualized embedding-based word aligner that extracts word alignments based on similarities of the tokens' contextualized embeddings. Awesome-align achieves state-of-the-art performance on five language pairs. Optionally, Awesome-align can be fine-tuned on parallel text with objectives suitable for word alignment and on gold alignment data.

We benchmark the word aligners on the gold standard alignments in the GALE Arabic–English Parallel Aligned Treebank (Li et al., 2012). We use the same data splits as Stengel-Eskin et al. (2019), containing 1687, 299, and 315 sentence pairs in the train, dev, and test splits, respectively. To obtain alignments using fast-align, we append the test

[13] Denormalization code available at https://github.com/KentonMurray/ArabicDetokenizer

Model	Layer†	AER	P	R	F
fast-align*	n/a	47.4	53.9	51.4	52.6
Awesome-align w/o FT					
mBERT	8	35.6	78.5	54.5	64.4
GBv4	8	**32.7**	**85.6**	55.4	**67.3**
XLM-R	16	40.1	78.6	48.4	59.9
L64K	17	34.0	81.5	**55.5**	66.0
L128K	17	35.1	80.0	54.5	64.9
Awesome-align w/ FT					
mBERT$_{ft}$	8	30.0	81.9	**61.2**	70.0
GBv4$_{ft}$	8	29.3	86.9	59.7	70.7
XLM-R$_{ft}$	18	**27.8**	**90.3**	60.2	**72.2**
L64K$_{ft}$	17	29.1	84.9	60.9	70.9
L128K$_{ft}$	16	32.2	80.3	58.7	67.8
Awesome-align w/ FT & supervision					
XLM-R$_{ft.s}$	16	**23.3**	92.5	**65.6**	**76.7**
L128K$_{ft.s}$	17	23.5	**93.7**	64.6	76.5

Table 3: Alignment performance on GALE EN–AR. *Trained on MT bitext. †We report the best layer of each encoder based on dev alignment error rate (AER).

data to the MT training bitext and run the tool from scratch. Awesome-align extracts the alignments for the test set based on pretrained contextualized embeddings. These encoders can be fine-tuned using the parallel text in the train and dev sets. Additionally, the encoders can be further fine-tuned using supervision from gold word alignments.

Intrinsic Evaluation Table 3 shows the performance of word alignment methods on the GALE English–Arabic alignment dataset. Awesome-align outperforms fast-align, and fine-tuned Awesome-align (*ft*) outperforms models that were not fine-tuned. Incorporating supervision from the gold alignments (*s*) leads to the best performance.

6 Cross-lingual Transfer Results

One might optimistically consider that the latest multilingual encoder (in this case XLM-R) in the zero-shot setting would achieve the best possible performance. However, in our extensive experiments in Table 4 and Table 5, we find that the zero-shot approach can usually be improved with data projection. In this section, we explore the impact of each factor within the data projection process.

6.1 English–Arabic Experiments

In Table 4, we present the Arabic test performance of five tasks under all combinations considered. The "MT" and "Align" columns indicate the models used for the translation and word alignment components of the data projection process. For ACE, we report results on the average of six metrics.[14] For a large bilingual encoder, we use L128K instead of L64K due to its slightly better performance on English ACE (Appendix E).

Impact of Data Projection By comparing any group against group Z, we observe adding silver data yields better or equal performance to zero-shot in at least some setup in the IE tasks (ACE, NER, and BETTER). For syntax-related tasks, we observe similar trends, with the exception of XLM-R. We hypothesize that XLM-R provides better syntactic cues than those obtainable from the alignment, which we discuss later in relation to self-training.

Impact of Word Aligner By comparing groups A, B, and C of the same encoder, we observe that Awesome-align performs overall better than statistical MT-based fast-align (FA). Additional fine-tuning (*ft*) on MT training bitext further improves its performance. As a result, we use fine-tuned aligners for further experiments. Moreover, incorporating supervised signals from gold alignments in the word alignment component (*ft.s*) often helps performance of the task. In terms of computation budget, these three groups use a publicly available MT system ("public"; Tiedemann, 2020) and require only fine-tuning the encoder for alignment, which requires small additional computation.

Impact of Encoder Size Large bilingual or multilingual encoders tend to perform better than base encoders in the zero-shot scenario, with the exception of the bilingual encoders on ACE and BETTER. While we observe base size encoders benefit from reducing the number of supported languages (from 100 to 2), for large size encoders trained much longer, the zero-shot performance of the bilingual model is worse than that of the multilingual model. After adding silver data from group C based on the public MT model and the fine-tuned aligner, the performance gap between base and large models tends to shrink, with the exception of both bilingual and multilingual encoders on NER. In terms of computation budget, training a bilingual encoder requires significant additional computation.

[14] Six metrics include entity, relation, trigger identification and classification, and argument identification and classification accuracies. See Appendix D for a breakdown of metrics.

	MT	Align	ACE	NER	POS	Parsing	BET.	ACE	NER	POS	Parsing	BET.
			mBERT (base, multilingual)					*XLM-R (large, multilingual)*				
(Z)	-	-	27.0	41.6	59.7	29.2	39.9	**45.1**	46.4	73.3	**48.0**	50.8
(A)	public	FA	+2.5	-3.8	+8.5	+7.3	+2.6	-7.5	-0.1	-7.7	-9.5	-1.6
(B)	public	mBERT	+6.5	+0.2	+8.5	+7.6	+2.3	-4.4	+6.9	-6.1	-8.4	-2.6
(B)	public	XLM-R	+0.9	-2.9	+9.5	+9.0	-1.2	-10.0	+0.0	-5.9	-8.8	-6.3
(C)	public	mBERT$_{ft}$	+7.8	**+5.6**	+7.7	+10.0	+4.1	-0.6	+7.4	-8.0	-6.8	+0.3
(C)	public	XLM-R$_{ft}$	+7.7	+4.9	+6.2	+9.3	+4.5	-2.6	+7.0	-9.0	-7.6	+1.0
(C)	public	XLM-R$_{ft.s}$	+7.3	+1.5	+10.1	**+12.4**	+4.8	-3.0	+9.1	-3.8	-3.7	**+2.3**
(D)	public	GBv4$_{ft}$	+8.5	+4.3	+5.9	+8.9	+5.0	-1.5	+7.7	-9.4	-9.1	-0.1
(D)	public	L128K$_{ft}$	+6.4	+3.1	+6.5	+8.2	+1.6	-1.6	+6.1	-9.0	-9.4	-3.6
(D)	public	L128K$_{ft.s}$	+7.0	+3.7	**+10.3**	+11.8	**+5.4**	-0.3	+5.2	-4.4	-4.6	+2.1
(E)	GBv4	mBERT$_{ft}$	+8.4	+3.2	+7.7	+9.9	+4.7	-1.5	+3.2	-7.1	-6.7	+0.7
(E)	GBv4	XLM-R$_{ft}$	+9.6	+1.8	+7.0	+9.5	+5.2	-0.4	+1.4	-8.3	-7.7	+1.4
(E)	L128K	mBERT$_{ft}$	**+12.1**	+3.3	+7.9	+9.9	+4.7	-1.4	+7.2	-8.1	-6.7	+1.3
(E)	L128K	XLM-R$_{ft}$	+10.2	-1.9	+6.1	+9.4	+4.8	-0.5	+4.6	-9.8	-7.5	+2.0
(S)	public	ST	-	+5.5	+0.1	-20.3	+0.3	-	**+10.0**	**+1.8**	-29.6	+1.2
			GBv4 (base, bilingual)					*L128K (large, bilingual)*				
(Z)	-	-	46.0	45.4	64.7	33.2	41.7	42.7	46.3	67.9	36.7	40.9
(C)	public	mBERT$_{ft}$	+0.6	+3.7	+2.6	+6.9	+7.5	+2.7	+8.2	-0.9	+4.9	+11.7
(C)	public	XLM-R$_{ft}$	-1.4	**+4.5**	+1.8	+6.0	+8.4	+1.2	**+9.0**	-2.5	+3.9	+10.5
(C)	public	XLM-R$_{ft.s}$	-0.1	+3.4	**+5.1**	**+9.2**	+8.0	+2.7	+7.0	+1.2	**+7.2**	**+12.1**
(E)	GBv4	mBERT$_{ft}$	-0.1	+0.1	+3.3	+7.2	+8.1	+4.2	-0.5	-0.1	+5.1	+11.2
(E)	GBv4	XLM-R$_{ft}$	+0.1	+0.4	+1.5	+6.0	**+9.7**	+2.4	+0.0	-1.3	+4.2	+10.8
(E)	L128K	mBERT$_{ft}$	-0.6	+1.0	+2.6	+6.1	+7.4	**+5.5**	+0.8	-0.7	+4.7	+10.6
(E)	L128K	XLM-R$_{ft}$	+0.9	-2.1	+1.1	+5.5	+7.8	+4.4	-3.6	-2.2	+4.1	+11.3
(F)	GBv4	GBv4$_{ft}$	+0.0	-1.9	+1.6	+4.5	+9.1	+2.0	-0.3	-1.7	+3.2	+10.9
(F)	GBv4	L128K$_{ft}$	-0.9	-1.4	+1.5	+4.1	+5.7	+2.3	-1.7	-2.4	+2.6	+8.3
(F)	L128K	GBv4$_{ft}$	-4.3	-1.0	+0.4	+4.1	+7.4	+4.1	-3.6	-2.1	+2.3	+11.4
(F)	L128K	L128K$_{ft}$	-3.5	-1.1	+0.3	+3.8	+ 4.5	+2.9	+0.1	-2.9	+2.0	+6.7
(F)	L128K	L128K$_{ft.s}$	**+1.9**	+0.2	+3.3	+7.4	+7.2	+2.8	-1.8	+0.8	+6.0	+11.8
(S)	public	ST	-	-2.5	-1.3	-18.6	+1.9	-	+7.1	**+1.5**	-21.7	+8.1

Table 4: Performance of Arabic on 5 tasks under various setups. Cells are colored by performance difference over zero-shot baseline: +5 or more , +1 to +5 , -1 to -5 , -5 or more . **Highlights** indicate the best setting for each task (best viewed in color). The best setting for each task and encoder combination is **bolded**. We order four encoders along two axes, similar to Table 1.

Impact of Encoder on Word Aligner By comparing groups C and D (in multilingual encoders) or groups E and F (in bilingual encoders), we observe bilingual encoders tend to perform slightly worse than multilingual encoders for word alignment. If bilingual encoders exist, using them in aligners requires little additional computation.

Impact of Encoder on MT By comparing groups C and E, we observe the performance difference between the bilingual encoder based MT and the public MT depends on the task and encoder, and neither MT system clearly outperforms the other in all settings, despite the bilingual encoder having a better BLEU score. The results suggest that both options should be explored if one's budget allows. In terms of computation budget, using pretrained encoders in a custom MT system requires medium additional computation.

Impact of Label Source To assess the quality of the projected annotations in the silver data, we consider a different way to automatically label translated sentences: self-training (ST; Yarowsky, 1995). For self-training, we translate the source data to the target language, label the translated data using a zero-shot model trained on source data, and combine the labeled translations with the source data to

Encoder	Data	ar	de	en	es	fr	hi	ru	vi	zh	Average
NER (F1)											
mBERT	Zero-shot	41.6	**78.8**	83.9	73.1	79.5	66.2	63.4	70.8	51.8	67.7
	+ Self	**+7.7**	-0.5	**+0.4**	+4.8	**+2.4**	-2.5	**+2.7**	**+1.2**	+1.4	**+2.0**
	+ Proj	-5.8	-0.6	+0.3	+3.6	+0.2	**+0.4**	-1.7	-2.0	**+2.3**	-0.4
	+ Proj (Bi)	+0.3	-0.7	+0.1	**+5.2**	-0.6	-2.1	-1.1	+0.3	+0.0	+0.2
XLM-R	Zero-shot	46.4	79.5	83.9	76.1	80.0	70.9	**70.5**	77.0	40.2	69.4
	+ Self	**+11.2**	**+0.9**	**+0.6**	**+1.0**	**+0.5**	+2.1	-1.5	**+1.7**	+2.3	**+2.1**
	+ Proj	+1.7	-0.7	-0.1	-3.9	-1.2	+1.2	-4.8	-9.1	+14.2	-0.3
	+ Proj (Bi)	+6.9	+0.4	-0.2	-4.3	-1.5	**+3.2**	-3.3	-5.2	**+15.1**	+1.2
POS (ACC)											
mBERT	Zero-shot	59.7	89.6	**96.9**	87.5	88.7	69.5	81.9	62.6	66.6	78.1
	+ Self	+0.3	**+0.5**	+0.0	**+0.4**	**+0.4**	-0.3	**+0.5**	**+0.4**	**+1.7**	**+0.4**
	+ Proj	+6.9	-3.2	+0.0	-3.8	-3.9	+1.3	-6.6	-7.4	-4.1	-2.3
	+ Proj (Bi)	**+8.5**	-2.6	-0.1	-3.2	-3.0	**+1.6**	-5.7	-6.9	-3.9	-1.7
XLM-R	Zero-shot	73.3	**91.5**	**98.0**	**89.3**	**90.0**	78.6	86.8	**65.2**	53.6	80.7
	+ Self	**+1.6**	-0.3	+0.0	+0.0	+0.0	**+2.0**	**+0.1**	-0.4	**+11.7**	**+1.6**
	+ Proj	-7.1	-5.4	-0.5	-6.3	-5.9	-6.0	-10.5	-8.9	+9.7	-4.6
	+ Proj (Bi)	-6.1	-4.6	-0.1	-4.9	-4.6	-5.5	-10.4	-8.7	+9.4	-4.0
Parsing (LAS)											
mBERT	Zero-shot	29.2	**67.7**	79.7	**68.9**	**73.2**	31.2	**60.6**	**33.6**	**29.4**	**52.6**
	+ Self	-20.6	-34.2	+0.1	-41.6	-41.1	-15.3	-35.2	-17.8	-14.5	-24.5
	+ Proj	**+9.1**	-2.1	**+1.1**	-4.9	-5.8	**+6.0**	-5.6	-7.2	-2.1	-1.3
	+ Proj (Bi)	+7.6	-1.6	+0.5	-3.8	-4.5	+5.7	-4.8	-7.2	-2.5	-1.2
XLM-R	Zero-shot	**48.0**	**69.6**	82.6	**73.6**	**76.1**	**43.1**	**70.3**	**38.4**	15.0	**57.4**
	+ Self	-30.4	-29.4	**+0.1**	-39.9	-40.0	-18.3	-33.9	-16.1	-9.7	-24.2
	+ Proj	-8.5	-4.3	+0.0	-10.3	-10.1	-5.7	-14.8	-11.1	+14.5	-5.6
	+ Proj (Bi)	-8.4	-1.6	**+0.1**	-7.7	-7.4	-3.1	-12.7	-9.8	**+15.1**	-3.9

Table 5: Performance of NER, POS, and parsing for eight target languages. We use the same color code as Table 4.

train a new model.[15] Compared to the silver data, the self-training data has the same underlying text but a different label source.

We first observe that self-training for parsing leads to significantly worse performance due to the low quality of the predicted trees. By comparing groups S and C, which use the same underlying text, we observe that data projection tends to perform better than self-training, with the exceptions of POS tagging with a large encoder and NER with a large multilingual encoder. These results suggest that the external knowledge[16] in the silver data complements the knowledge obtainable when the model is trained with source language data alone, but when the zero-shot model is already quite good (like for POS tagging) data projection can harm performance compared to self-training. Future directions could include developing task-specific projection and alignment heuristics to improve projected annotation quality for POS tagging or parsing, and combining data projection and self-training.

6.2 Multilingual Experiments

In Table 5, we present the test performance of three tasks for eight target languages. We use the public MT system (Tiedemann, 2020) and non-fine-tuned Awesome-align with mBERT as the word aligner for data projection—a setup with the smallest computation budget—due to computation constraints. We consider both data projection (+Proj) and self training (+Self). We use silver data in addition to English gold data for training. We use multilingual training with +Self and +Proj, and bilingual training with +Proj (Bi).

We observe that data projection (+Proj (Bi)) sometimes benefits languages with the lowest zero-shot performance (Arabic, Hindi, and Chinese), with the notable exception of XLM-R on syntax-based tasks (excluding Chinese). For languages closely related to English, data projection tends to

[15] This setup differs from traditional zero-shot self-training in cross-lingual transfer, as the traditional setup assumes unlabeled corpora in the target language(s) (Eisenschlos et al., 2019) instead of translations of the source language data.

[16] "External knowledge" refers to knowledge introduced into the downstream model as a consequence of the particular decisions made by the aligner (and subsequent projection).

hurt performance. We observe that for data projection, training multiple bilingual models (+Proj (Bi)) outperforms joint multilingual training (+Proj). This could be the result of noise from alignments of various quality mutually interfering. In fact, self-training with the same translated text (+Self) outperforms data projection and zero-shot scenarios, again with the exception of parsing. As data projection and self-training use the same translated text and differ only by label source, the results indicate that the external knowledge from frozen mBERT-based alignment is worse than what the model learns from source language data alone. Thus, further performance improvement could be achieved with an improved aligner.

7 Related Work

Although projected data may be of lower quality than the original source data due to errors in translation or alignment, it is useful for tasks such as semantic role labeling (Akbik et al., 2015; Aminian et al., 2019), information extraction (Riloff et al., 2002), POS tagging (Yarowsky and Ngai, 2001), and dependency parsing (Ozaki et al., 2021). The intuition is that although the projected data may be noisy, training on it gives a model useful information about the statistics of the target language.

Akbik et al. (2015) and Aminian et al. (2017) use bootstrapping algorithms to iteratively construct projected datasets for semantic role labeling. Akbik et al. (2015) additionally use manually defined filters to maintain high data quality, which results in a projected dataset that has low recall with respect to the source corpus. Fei et al. (2020) and Daza and Frank (2020) find that a non-bootstrapped approach works well for cross-lingual SRL. Advances in translation and alignment quality allow us to avoid bootstrapping while still constructing projected data that is useful for downstream tasks.

Fei et al. (2020) and Daza and Frank (2020) also find improvements when training on a mixture of gold source language data and projected silver target language data. Ideas from domain adaptation can be used to make more effective use of gold and silver data to mitigate the effects of language shift (Xu et al., 2021).

Improvements to task-specific models for zero-shot transfer are orthogonal to our work. For example, language-specific information can be incorporated using language indicators or embeddings (Johnson et al., 2017), contextual parameter generators (Platanios et al., 2018), or language-specific semantic spaces (Luo et al., 2021). Conversely, adversarial training (Ganin et al., 2016) has been used to discourage models from learning language-specific information (Chen et al., 2018; Keung et al., 2019; Ahmad et al., 2019).

8 Conclusion

In this paper, we explore data projection and the use of silver data in zero-shot cross-lingual IE, facilitated by neural machine translation and word alignment. Recent advances in pretrained encoders have improved machine translation systems and word aligners in terms of intrinsic evaluation. We conduct an extensive extrinsic evaluation and study how the encoders themselves—and components containing them—impact performance on a range of downstream tasks and languages.

With a test bed of English–Arabic IE tasks, we find that adding projected silver training data overall yields improvements over zero-shot learning. Comparisons of how each factor in the data projection process impacts performance show that while one might hope for the existence of a silver bullet strategy, the best setup is usually task dependent.

In multilingual experiments, we find that silver data tends to help languages with the weakest zero-shot performance, and that it is best used separately for each desired language pair instead of in joint multilingual training.

We also examine self-training with translated text to assess when data projection helps cross-lingual transfer, and find it to be another viable option for obtaining labels for some tasks. In future work, we will explore how to improve alignment quality and how to combine data projection and self-training techniques.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. We thank João Sedoc for helpful discussions and Shabnam Behzad for post-submission experiments. This work was supported in part by IARPA BETTER (#2019-19051600005). The views and conclusions contained in this work are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, or endorsements of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes

notwithstanding any copyright annotation therein.

This research made use of the following open-source software: AllenNLP (Gardner et al., 2018), FairSeq (Ott et al., 2019), NumPy (Harris et al., 2020), PyTorch (Paszke et al., 2017), PyTorch lightning (Falcon, 2019), scikit-learn (Pedregosa et al., 2011), and Transformers (Wolf et al., 2020).

References

Wasi Uddin Ahmad, Zhisong Zhang, Xuezhe Ma, Kai-Wei Chang, and Nanyun Peng. 2019. Cross-lingual dependency parsing with unlabeled auxiliary languages. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 372–382, Hong Kong, China. Association for Computational Linguistics.

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015. Generating high quality proposition Banks for multilingual semantic role labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 397–407, Beijing, China. Association for Computational Linguistics.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2017. Transferring semantic roles using translation and syntactic information. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 13–19, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2019. Cross-lingual transfer of semantic roles: From raw text to semantic roles. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 200–210, Gothenburg, Sweden. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2753–2765, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

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*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Angel Daza and Anette Frank. 2020. X-SRL: A parallel cross-lingual semantic role labeling dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3904–3914, Online. Association for Computational Linguistics.

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

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger, and Jeremy Howard. 2019. MultiFiT: Efficient multi-lingual language model fine-tuning. 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 5702–5707, Hong Kong, China. Association for Computational Linguistics.

Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *arXiv preprint arXiv:1611.04033*.

WA Falcon. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3.

Hao Fei, Meishan Zhang, and Donghong Ji. 2020. Cross-lingual semantic role labeling with high-quality translated training corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7014–7026, Online. Association for Computational Linguistics.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

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

Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585:357–362.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

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 *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Phillip Keung, Yichao Lu, and Vikas Bhardwaj. 2019. Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and NER. 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 1355–1360, Hong Kong, China. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. An empirical study of pre-trained transformers for Arabic information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4727–4734, Online. Association for Computational Linguistics.

Xuansong Li, Stephanie Strassel, Stephen Grimes, Safa Ismael, Mohamed Maamouri, Ann Bies, and Nianwen Xue. 2012. Parallel aligned treebanks at LDC: New challenges interfacing existing infrastructures. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1848–1855, Istanbul, Turkey. European Language Resources Association (ELRA).

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. 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. *arXiv preprint arXiv:1907.11692*.

Shengjie Luo, Kaiyuan Gao, Shuxin Zheng, Guolin Ke, Di He, Liwei Wang, and Tie-Yan Liu. 2021. Revisiting language encoding in learning multilingual representations. *arXiv preprint arXiv:2102.08357*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Hiroaki Ozaki, Gaku Morio, Terufumi Morishita, and Toshinori Miyoshi. 2021. Project-then-transfer: Effective two-stage cross-lingual transfer for semantic dependency parsing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2586–2594, Online. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Rrubaa Panchendrarajan and Aravindh Amaresan. 2018. Bidirectional LSTM-CRF for named entity recognition. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Robert Parker, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda. 2011. Arabic gigaword fifth edition ldc2011t11. *Philadelphia: Linguistic Data Consortium*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 425–435, Brussels, Belgium. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Prokopis Prokopidis, Vassilis Papavassiliou, and Stelios Piperidis. 2016. Parallel Global Voices: a collection of multilingual corpora with citizen media stories. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 900–905, Portorož, Slovenia. European Language Resources Association (ELRA).

Ellen Riloff, Charles Schafer, and David Yarowsky. 2002. Inducing information extraction systems for new languages via cross-language projection. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Sebastian Ruder, Noah Constant, Jan A. Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Graham Neubig, and Melvin Johnson. 2021. Xtreme-r: Towards more challenging and nuanced multilingual evaluation. *ArXiv*, abs/2104.07412.

Hassan Sajjad, Francisco Guzmn, Preslav Nakov, Ahmed Abdelali, Kenton Murray, Fahad Al Obaidli, and Stephan Vogel. 2013. Qcri at iwslt 2013: Experiments in arabic-english and english-arabic spoken language translation. In *Proceedings of the 10th International Workshop on Spoken Language Technology (IWSLT-13)*. Citeseer.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Elias Stengel-Eskin, Tzu-ray Su, Matt Post, and Benjamin Van Durme. 2019. A discriminative neural model for cross-lingual word alignment. 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 910–920, Hong Kong, China. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014. Treebank translation for cross-lingual parser induction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 130–140, Ann Arbor, Michigan. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus LDC2006T06. *Web Download. Philadelphia: Linguistic Data Consortium*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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.

Shijie Wu and Mark Dredze. 2020a. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020b. Do explicit alignments robustly improve multilingual encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4471–4482, Online. Association for Computational Linguistics.

Patrick Xia, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme. 2021. LOME: Large ontology multilingual extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 149–159, Online. Association for Computational Linguistics.

Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray. 2021. Gradual fine-tuning for low-resource domain adaptation. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 214–221, Kyiv, Ukraine. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*.

Daniel Zeman et al. 2020. Universal dependencies 2.7. *LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University*.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

A Fine-tuning Hyperparameters

A.1 ACE

We used the OneIE v0.4.8 codebase[17] with the following hyperparameters: Adam optimizer (Kingma and Ba, 2014) for 60 epochs with a learning rate of 5e-5 and weight decay of 1e-5 for the encoder, and a learning rate of 1e-3 and weight decay of 1e-3 for other parameters. Two-layer feed-forward network with a dropout rate of 0.4 for task-specific classifiers, 150 hidden units for entity and relation extraction, and 600 hidden units for event extraction. β_v and β_e set to 2 and θ set to 10 for global features. Data statistics can be found in Table 6.

	Train (en/ar*)	Dev (en/ar*)	Test (en/ar)
Sent.	19,216/1,710	901/256	676/216
Evt. trig.	4,419/1,825	468/211	424/234
Evt. arg.	6,607/3,255	759/412	689/451
Entity	47,554/25,889	3,423/3,554	3,673/2,977
Relation	7,159/3,704	728/527	802/478
Rel. arg.	14,318/7,408	1,456/1,054	1,604/956

Table 6: ACE dataset statistics. *Arabic train and dev sets are not used in our experiments.

	Train (en)	Dev (en)	Test (ar)
Sent.	3,629	453	129
Evt. trig.	12,390	1,527	517
Evt. arg.	20,314	2,522	857

Table 7: BETTER dataset statistics.

A.2 BETTER

The codebase for event structure prediction uses AllenNLP (Gardner et al., 2018). The contextual encoder produces representations for the tagger and typer modules. Span representations are formed by concatenating the output of a self-attention layer over the span's token embeddings with the embeddings of the first and last tokens of the span. The BiLSTM-CRF tagger has 2 layers, both with hidden size of 2048. We use a dropout rate of 0.3 and maximum sequence length of 512. Child span prediction is conditioned on parent spans and labels, so we represent parent labels with an embedding of size 128. We use Adam optimizer to fine-tune the encoder with a learning rate of 2e-5, and we use a learning rate of 1e-3 for other components. The tagger loss is negative log likelihood and the typer loss is cross entropy. We equally weight both losses and train against their sum. The contextual encoder is not frozen. Data statistics can be found in Table 7.

[17] http://blender.cs.illinois.edu/software/oneie/

A.3 NER, POS Tagging, and Parsing

We use the Adam optimizer with a learning rate of 2e-5 with linear warmup for the first 10% of total steps and linear decay afterwards, and train for 5 epochs with a batch size of 32. To obtain valid BIO sequences, we rewrite standalone `I-X` into `B-X` and `B-X I-Y I-Z` into `B-Z I-Z I-Z`, following the final entity type. For parsing, we ignore punctuations (PUNCT) and symbols (SYM) when calculating LAS.

We set the maximum sequence length to 128 during fine-tuning. For NER and POS tagging, we additionally use a sliding window of context to include subwords beyond the first 128. At test time, we use the same maximum sequence length except for parsing. At test time for parsing, we use only the first 128 words of a sentence. As the supervision for Chinese NER is character-level, we segment the characters into words using the Stanford Word Segmenter and realign the label.

The datasets we used are publicly available: NER,[18] POS tagging, and dependency parsing.[19] Data statistics can be found in Table 8.

	NER	POS tagging Parsing
en-train	20,000	12,543
en-dev	10,000	2,002
en-test	10,000	2,077
ar-test	10,000	680
de-test	10,000	977
es-test	10,000	426
fr-test	10,000	416
hi-test	1,000	1,684
ru-test	10,000	601
vi-test	10,000	800
zh-test	10,000	500

Table 8: Number of examples.

B Encoder Pretraining Hyperparameters

We pretrain each encoder with a batch size of 2048 sequences and 512 sequence length for 250K steps

[18] https://www.amazon.com/clouddrive/share/d3KGCRCIYwhKJF0H3eWA26hjg2ZCRhjpEQtDL70FSBN

[19] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3424

from scratch,[20] roughly 1/24 the amount of pretraining compute of XLM-R. Training takes 8 RTX 6000 GPUs roughly three weeks. We follow the pretraining recipe of RoBERTa (Liu et al., 2019) and XLM-R. We omit the next sentence prediction task and use a learning rate of 2e-4, Adam optimizer, and linear warmup of 10K steps then decay linearly to 0, multilingual sampling alpha of 0.3, and the fairseq (Ott et al., 2019) implementation.

C Machine Translation Hyperparameters

All of our machine translation systems are based upon the Transformer architecture: a six-layer encoder, six-layer decoder model with 2048 FFN dimension and 8 attention heads. We use 4 Nvidia V100 GPUs, with a batch size of 2048 tokens per GPU. We accumulate the gradient 10 times before updating model parameters. The initial learning rate is 1e-3. The optimizer is Adam with an `inverse_sqrt` learning rate scheduler. In the inference step, the width of beam search is 4 with a length penalty of 0.6.

D ACE Arabic Full Metrics

The full metrics of Arabic ACE can be found in Table 9 and Table 10.

E ACE English Full Metrics

The full metrics of English ACE can be found in Table 11.

[20] While we use XLM-R as the initialization of the Transformer, due to vocabulary differences, the learning curve is similar to that of pretraining from scratch.

	MT	Align	Entity	Relation	Trig-I	Trig-C	Arg-I	Arg-C	AVG
mBERT (base, multilingual)									
(Z)	-	-	59.3	25.7	23.8	22.2	17.2	13.8	27.0
(A)	public	FA	-2.2	-13.9	+6.5	+2.5	+10.7	+11.5	+2.5
(B)	public	mBERT	-6.2	-5.1	+16.0	+10.6	+11.5	+12.1	+6.5
(B)	public	XLM-R	-12.7	-17.9	+11.1	+8.0	+8.5	+8.1	+0.9
(C)	public	mBERT$_{ft}$	-1.1	+0.9	+12.8	+9.8	+10.9	+13.6	+7.8
(C)	public	XLM-R$_{ft}$	-0.1	-4.2	+16.0	+11.9	+11.2	+11.3	+7.7
(C)	public	XLM-R$_{ft.s}$	-0.2	-1.6	+13.4	+11.5	+9.0	+11.7	+7.3
(D)	public	GBv4$_{ft}$	-1.9	+2.8	+14.3	+9.9	+12.7	+13.3	+8.5
(D)	public	L128K$_{ft}$	-1.7	+0.6	+11.6	+8.3	+10.7	+9.0	+6.4
(D)	public	L128K$_{ft.s}$	-1.3	+3.6	+12.7	+8.4	+8.3	+10.3	+7.0
(E)	GBv4	mBERT$_{ft}$	+1.0	+4.7	+13.6	+10.3	+9.3	+11.3	+8.4
(E)	GBv4	XLM-R$_{ft}$	-0.5	+5.5	+12.6	+10.8	+15.1	+14.4	+9.6
(E)	L128K	mBERT$_{ft}$	+2.6	+5.2	+12.9	+13.4	+18.8	+19.6	+12.1
(E)	L128K	XLM-R$_{ft}$	+2.5	+6.3	+11.2	+5.1	+17.1	+19.2	+10.2
XLM-R (large, multilingual)									
(Z)	-	-	70.0	38.7	44.0	40.8	39.5	37.8	45.1
(A)	public	FA	-7.2	-9.5	-9.3	-8.2	-4.8	-6.0	-7.5
(B)	public	mBERT	-8.5	-10.2	-2.2	-0.1	-2.0	-3.4	-4.4
(B)	public	XLM-R	-14.7	-12.1	-8.9	-7.8	-8.1	-8.3	-10.0
(C)	public	mBERT$_{ft}$	-2.5	+3.5	-2.3	-4.3	+1.8	+0.2	-0.6
(C)	public	XLM-R$_{ft}$	-3.8	-0.5	-3.6	-4.5	-0.5	-2.8	-2.6
(C)	public	XLM-R$_{ft.s}$	-2.4	+1	-3.6	-7.5	-2.1	-3.2	-3.0
(D)	public	GBv4$_{ft}$	-4.4	+0.8	+0.8	-2.3	-1.1	-2.8	-1.5
(D)	public	L128K$_{ft}$	-2.1	-1.4	+0.6	-2.2	-2.0	-3.0	-1.6
(D)	public	L128K$_{ft.s}$	-0.9	+2.2	-1.6	-4.4	+1.8	+0.8	-0.3
(E)	GBv4	mBERT$_{ft}$	-2.5	+1.4	-3.2	-4.1	+0.3	-0.9	-1.5
(E)	GBv4	XLM-R$_{ft}$	-2.2	+2.3	-2.2	-3.0	+2.9	-0.3	-0.4
(E)	L128K	mBERT$_{ft}$	+0.1	-1.1	-2.2	-3.0	-0.9	-1.3	-1.4
(E)	L128K	XLM-R$_{ft}$	+0.1	+4.2	-4.4	-6.3	+1.8	+1.3	-0.5
GBv4 (base, bilingual)									
(Z)	-	-	71.9	29.6	49.8	46.8	41.1	36.8	46.0
(C)	public	mBERT$_{ft}$	-0.1	+9.3	-5.8	-5.8	+3.0	+3.0	+0.6
(C)	public	XLM-R$_{ft}$	-0.8	+10.4	-8.0	-9.0	-1.5	+0.6	-1.4
(C)	public	XLM-R$_{ft.s}$	+0.0	+9.7	-8.0	-7.0	+1.7	+2.9	-0.1
(E)	GBv4	mBERT_FT	-2.0	+7.6	-4.9	-5.8	+1.8	+2.5	-0.1
(E)	GBv4	XLMR_FT	-1.1	+9.8	-8.5	-7.7	+3.8	+4.2	+0.1
(E)	L128K	mBERT_FT	-3.0	+8.1	-4.9	-3.9	-0.8	+0.9	-0.6
(E)	L128K	XLMR_FT	-0.4	+11.6	-5.8	-4.9	+1.4	+3.4	+0.9
(F)	GBv4	GBv4$_{ft}$	-1.9	+8.1	-4.7	-4.6	+1.7	+1.7	+0.0
(F)	GBv4	L128K$_{ft}$	-0.5	+5.4	-5.1	-5.2	-0.6	+0.6	-0.9
(F)	L128K	GBv4$_{ft}$	-0.6	+7.6	-4.5	-5.4	-0.2	+0.3	-4.3
(F)	L128K	L128K$_{ft}$	-1.1	+8.6	-6.6	-4.7	+4.2	+5.8	-3.5
(F)	L128K	L128K$_{ft.s}$	+0.0	+10.5	-4.6	-5.6	+4.9	+6.0	+1.9

Table 9: Detailed performance of bilingual English–Arabic ACE. Cells are colored following Table 4.

	MT	Align	Entity	Relation	Trig-I	Trig-C	Arg-I	Arg-C	AVG
L128K (large, bilingual)									
(Z)	-	-	66.0	30.7	44.0	43.0	37.4	35.4	42.7
(C)	public	mBERT$_{ft}$	+2.1	+7.8	+1.8	-0.6	+3.1	+1.6	+2.7
(C)	public	XLM-R$_{ft}$	+2.6	+5.2	-1.7	-4.2	+3.2	+1.9	+1.2
(C)	public	XLM-R$_{ft.s}$	+5.7	+8.9	-2.6	-4.2	+4.6	+4.0	+2.7
(E)	GBv4	mBERT_FT	+3.8	+12.7	+2.4	-0.1	+3.6	+2.3	+4.2
(E)	GBv4	XLMR_FT	+2.6	+11.9	-2.6	-5.1	+3.7	+3.4	+2.4
(E)	L128K	mBERT_FT	+3.8	+8.3	+2.2	+0.6	+8.9	+9.0	+5.5
(E)	L128K	XLMR_FT	+2.8	+9.6	+3.5	+0.8	+4.8	+4.8	+4.4
(F)	GBv4	GBv4$_{ft}$	+1.9	+6.7	+1.3	-1.5	+2.0	+1.6	+2.0
(F)	GBv4	L128K$_{ft}$	+2.7	+7.1	-1.1	-3.5	+4.9	+3.1	+2.3
(F)	L128K	GBv4$_{ft}$	+2.2	+8.9	+2.3	-0.2	+6.7	+4.1	+4.1
(F)	L128K	L128K$_{ft}$	+3.5	+7.0	-0.3	-2.1	+4.9	+3.9	+2.9
(F)	L128K	L128K$_{ft.s}$	+3.6	+11.7	-1.1	-3.7	+3.3	+2.9	+2.8

Table 10: Detailed performance of bilingual English–Arabic ACE. Cells are colored following Table 4.

Model	Train	Test	Entity	Relation	Trig-I	Trig-C	Arg-I	Arg-C	AVG
Lin et al. (2020)	en	en	89.6	58.6	75.6	72.8	57.3	54.8	68.1
BERT$_{large}$	en	en	90.2	64.0	75.7	73.2	59.5	57.4	70.0
mBERT	en	en	89.5	56.7	72.4	69.2	53.3	50.5	65.3
GBv4	en	en	90.2	63.0	73.8	71.4	57.7	55.4	68.6
XLM-R	en	en	90.9	**64.4**	75.3	72.2	58.4	55.5	69.4
L64K	en	en	91.30	64.0	**75.45**	73.0	59.4	57.4	70.1
L128K	en	en	**91.32**	64.1	75.39	**73.5**	**59.6**	**57.7**	**70.3**

Table 11: ACE results with different encoders. All models are trained and tested on gold English data.