Frustratingly Easy Label Projection for Cross-lingual Transfer

Yang Chen, Chao Jiang, Alan Ritter, Wei Xu
Georgia Institute of Technology
{yang.chen, chao.jiang, alan.ritter, wei.xu}@cc.gatech.edu

Abstract

Translating training data into many languages has emerged as a practical solution for improving cross-lingual transfer. For tasks that involve span-level annotations, such as information extraction or question answering, an additional label projection step is required to map annotated spans onto the translated texts. Recently, a few efforts have utilized a simple mark-then-translate method to jointly perform translation and projection by inserting special markers around the labeled spans in the original sentence (Lewis et al., 2020; Hu et al., 2020). However, as far as we are aware, no empirical analysis has been conducted on how this approach compares to traditional annotation projection based on word alignment. In this paper, we present an extensive empirical study across 57 languages and three tasks (QA, NER, and Event Extraction) to evaluate the effectiveness and limitations of both methods, filling an important gap in the literature. Experimental results show that our optimized version of mark-then-translate, which we call EasyProject, is easily applied to many languages and works surprisingly well, outperforming the more complex word alignment-based methods. We analyze several key factors that affect the end-task performance, and show EasyProject works well because it can accurately preserve label span boundaries after translation. [1]

1 Introduction

Zero-shot cross-lingual transfer, where models trained on a source language (e.g., English) are directly applied to other target languages, has the potential to extend NLP systems to many languages (Nooralahzadeh et al., 2020; Keung et al., 2020; Chen and Ritter, 2021; Niu et al., 2022; Huang et al., 2022a). Yet, its performance still lags behind models that are directly fine-tuned on labeled data (if available) from the target language. Recent work has shown that combining training data in a source language together with its automatic translation to the target language leads to consistent performance improvements (Xue et al., 2021; Hu et al., 2020). However, for NLP tasks that involve span-level annotations, an additional label projection step is needed to map the span annotations onto the translated texts (see Figure 1).

Traditionally, this annotation projection step is performed based on word alignment after machine translation (Akbik et al., 2015a; Aminian et al., 2019). To avoid the use of complex word alignment models, several recent efforts (Lewis et al., 2020; Hu et al., 2020) directly translated sentences with span annotations wrapped between special markers (e.g., <a> and). However, due to limited analysis presented in prior work, it is unclear (1) how well this approach works across different language families, (2) how robust MT systems are in handling special markers, as inserting markers inevitably degrades the translation quality, and (3) how well marker-based projection works in comparison to traditional alignment-based methods.

In this paper, we present the first systematic study of the mark-then-translate annotation projection technique, which includes careful evaluation of the choice of markers, projection accuracy, impact on translation quality, robustness to different MT systems, as well as a comparison to traditional alignment-based method across 57 languages (including 18 from Africa) on 5 datasets and 3 NLP tasks. We also propose an improved variant of marker-based projection, EASYPROJECT, that consistently outperforms the alignment-based approach, while being incredibly easy to use to project a variety of annotations (QA, entities,

[1] Our code and data is available at: https://github.com/edchengg/easyproject

Findings of the Association for Computational Linguistics: ACL 2023, pages 5775–5796
July 9-14, 2023

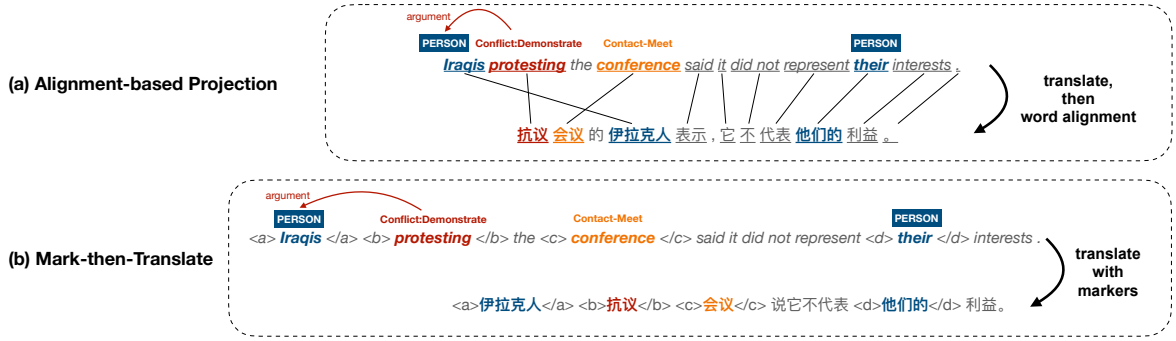

Figure 1: Two methods for translating and projecting English ACE event triggers and named entities to Chinese. (a) Pipeline method based on word alignment: starting with machine translation of the English sentence to Chinese, followed by word-to-word alignment. Then, labeled spans are projected using heuristics. (b) Mark-then-translate: markers are inserted around the annotated spans in the text. The modified sentence with markers inserted is then fed as input to an MT system, projecting the label span markers to the target sentence as a byproduct of translation.

relations, events) across many languages. The key is to use language-agnostic square bracket markers, combined with an efficient fine-tuning strategy to encourage the multilingual MT system to better preserve the special markers during translation.

Our main findings include (1) the marker-based method is surprisingly robust across different translation systems and languages, but the choice of markers matters (§3.2); (2) EasyProject can project annotated spans more accurately and is better at preserving span boundaries than the alignment-based methods, which is key to its success (§5.1); (3) fine-tuning an MT system for only 200 steps is sufficient to improve its robustness in handling special markers during translation (§4); (4) the margin of improved cross-lingual transfer is related to the language/script family and amount of pre-training data included in the multilingual model (§5.2). We hope our work will inspire more research on robust models that better handle text markup for the purpose of generating span annotations.

2 Background and Related Work

Alignment-based Projection. Projecting annotations via word alignment typically consists of the following steps: machine translate the available training data into the target language; run word alignment tools on the original and translated sentences; and finally, apply heuristics to map the span-level annotations from the original to translated texts. Statistical word alignment tools such as GIZA++ (Och and Ney, 2003) and fast-align (Dyer et al., 2013) have been widely adopted for projecting part-of-speech tags (Yarowsky et al., 2001; Eskander et al., 2020), semantic roles (Akbik et al., 2015b; Aminian et al., 2017; Daza and Frank, 2020; Fei et al., 2020), slot filling (Xu et al., 2020), semantic parser (Moradshahi et al., 2020; Nicosia et al., 2021), and NER labels (Ni et al., 2017; Stengel-Eskin et al., 2019). Recent progress on supervised neural aligners (Jalili Sabet et al., 2020; Nagata et al., 2020; Dou and Neubig, 2021; Lan et al., 2021) and multilingual contextualized embeddings (Devlin, 2018; Conneau et al., 2020) has further improved alignment accuracy. However, this pipeline-based method suffers from error propagation, translation shift (Akbik et al., 2015b), and non-contiguous alignments (Zenkel et al., 2020). Our analysis in §5.1 shows that the alignment-based methods are more error-prone when projecting span-level annotations, compared to the marker-based approaches.

Marker-based Projection. A few efforts have used mark-then-translate label projection method to translate question answering datasets into other languages (Lee et al., 2018; Lewis et al., 2020; Hu et al., 2020; Bornea et al., 2021). However, the focus of these papers was not the label projection task itself and there was no in-depth analysis on the effectiveness of the approach. For instance, Lewis et al. (2020) used quotation marks to translate the SQuAD training set into other languages but did not present empirical comparison to any other label projection methods. Similarly, Hu et al. (2020) used XML tags for the same purpose when creating the XTREME dataset, but this was only briefly mentioned in a few sentences in appendix. Besides QA, MulDA (Liu et al., 2021; Zhou et al., 2022) is a labeled sequence translation method that replaces entities with variable names for cross-lingual NER.

However, no comparison with existing projection methods was presented, as the main focus is generating synthetic labeled data using language models.

3 Analysis of Marker-Based Projection

The idea of marker-based label projection is straightforward – wrap labeled spans with special marker tokens, then translate the modified sentence (see an example in Figure 1b). The projected spans can be directly decoded from the translation if the markers are retrained. However, inserting markers inevitably degrades the translation quality. In this section, we analyze several questions left open by prior work (Lewis et al., 2020; Hu et al., 2020), including (1) how well are the special markers being preserved in translation, (2) the impact of different marker choices on translation quality and the performance of cross-lingual transfer.

3.1 Experimental Setup

We conduct experiments on three NLP tasks and 57 languages with five multilingual datasets to comprehensively evaluate the marker-based method. Most multilingual datasets are created by either (1) directly collecting and annotating data in the target language, or (2) translating English data with human or machine translation and then projecting labels manually or automatically. Four of our selected datasets were created with the first method, as evaluation on the translated datasets may overestimate performance on a target language, when in fact a model might only perform well on *translationese* (Riley et al., 2020).

Datasets. Our experiments include NER via the WikiANN (Pan et al., 2017; Rahimi et al., 2019) and MasahkaNER 2.0 (Adelani et al., 2022) datasets (§5.1), in addition to CoNLL-2002/2003 multilingual NER datasets (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) for comparison with Liu et al. (2021) (§F.1). For event extraction, we use the ACE05 corpus (Walker et al., 2006), which consists of six sub-tasks: entity and relation extraction, event trigger/argument identification and classification. For QA, we use TyDiQA-GoldP (Clark et al., 2020), which contains challenging questions written in eight languages. Data statistics are shown in Table 1 and 10 in Appendix.

	WikiANN	MasakhaNER	ACE05	TyDiQA
# of Lang.	39	20 (from Africa)	2	8
# of Docs	–	–	526/31/40	3,696/440/–
# of Sent.	20k/10k/10k	4.4k/638/1.2k	19k/901/676	17k/2,122/–
Avg. Length	–/8.0	–/23.9	519.3/14.9	96.8/21.0
Avg. # of Spans	1.4	1.8	2.9	1.0

Table 1: The detailed statistics of train/dev/test sets for each dataset. **Avg. Length** represents the average number of tokens in each article/sentence, and **Avg. # of Spans** denotes the average number of annotated spans in each sentence (in each article for TyDiQA).

IE and QA Models. We use XLM-RoBERTa$_{\text{large}}$ (Conneau et al., 2020) as the backbone model, except where noted.[2] For NER and QA, we fine-tune XLM-R with standard tagging and SQuAD-style span prediction layers. For event extraction, we use the OneIE framework (Lin et al., 2020), a joint neural model for information extraction with global features. We report average F_1 scores over three runs with different random seeds. More implementation details can be found in Appendix D.

MT Systems. We experiment with two MT systems: (1) the Google Translation (GMT) API,[3] and (2) an open-sourced multilingual translation model NLLB (No Language Left Behind) (Costa-jussà et al., 2022) with 3.3 billion parameters, supporting the translation between any pair of 200 languages.[4]

3.2 Choice of Markers

Ideally, a good span marker should minimize the impact on translation quality while having a high chance of being preserved during translation. However, prior works used quotation marks (“ ”) (Lee et al., 2018; Lewis et al., 2020) and XML/HTML tags (e.g., <a> or <PER>) (Hu et al., 2020; Ahmad et al., 2021) without much justification, which we address below.

Preserved in Translation. In a pilot study, we experimented with several markers, including XML tags, `[]`, “ ”, `()`, <>, and {}, etc. We found that both MT systems work reasonably well to retain square brackets (`[]`) and XML markers during the translation across many languages, while other markers that have language-specific formats

[2] We also experiment with mT5$_{large}$, mT5$_{\text{XL}}$, and mT5$_{\text{XXL}}$ (Xue et al., 2021) on WikiANN-NER (Table 7).

[3] https://cloud.google.com/translate

[4] https://github.com/facebookresearch/fairseq/tree/nllb

are easily lost in translation. For example, quotation marks (“ ”) are often translated in a language-specific way, e.g., «» in Russian, and are sometimes lost entirely in Arabic and Finnish, leading to low projection rates: 53% for Russian, 76% for Arabic, and 79% for Finnish based on TyDiQA dataset. The *projection rate* is measured by the percentage of data in which the numbers and type of special markers in the translations match with the source sentences. To improve the robustness of MT system in handling markers, we found further fine-tuning the MT system on synthetic data, where the special markers are inserted around name entities in parallel sentences, for only 200 steps is sufficient to boost the projection rate while maintaining translation quality (more details in §4).

Impact on Translation Quality. After narrowing down the choices to XML tags and square brackets, we further measure the impact of adding markers on the translation quality by adopting the evaluation setup used by Fan et al. (2021). We compare translation quality, with and without markers inserted, from English to various target languages using BLEU score. Table 2 presents the experimental results with Google Translation. Examples of errors are shown in Table 3. We find that inserting special markers indeed degrades translation quality, but overall, square brackets have less negative impact compared to XML tags. We hypothesize this is because using [] introduces less number of extra subword tokens in the encoding and decoding of the text during translation, compared to XML tags. More results on 55 languages using the NLLB translation system and more details about the evaluation setup can be found in Appendix G.1.

en → **Lang.**	**Corpus**	**# sent**	GMT - **BLEU**		
			Orig.	XML	[]
Arabic (*ar*)	TED18	1,997	**20.7**	14.0	15.1
German (*de*)	TED18	1,997	**44.5**	33.9	41.9
Spanish (*es*)	TED18	1,997	**45.9**	34.2	35.4
French (*fr*)	TED18	1,997	**37.6**	31.0	31.9
Hindi (*hi*)	TED18	1,070	**14.5**	12.8	13.0
Russian (*ru*)	WMT	1,997	**36.4**	28.5	35.2
Vietnamese (*vi*)	TED18	1,997	**32.8**	28.5	27.0
Chinese (*zh*)	WMT	1,997	**40.6**	33.4	37.1
AVG		1,881	**34.1**	27.0	29.6

Table 2: Comparsion of translation quality with different span markers, where the **best** and second best are marked. Overall, square brackets ([]) have less negative impact compared to XML tags. “Orig.” denotes the translation when no marker is inserted.

English #1:	The divorce settlement called for Giuliani to pay Hanover more than $6.8 million, according to the reporter .
Orig.:	据记者称，离婚协议要求朱利安尼向汉诺威支付超过680万美元。
[]:	据[记者]报道，[离婚]协议要求[朱利安尼][支付][汉诺威]超过680万美元。
XML:	据<e>记者<，<a>离婚和解协议要求朱利安尼<c>支付</c><d>汉诺威</d>超过680万美元。/e> 。
English #2:	The WTO is headquartered in Geneva .
Orig.:	يقع المقر الرئيسي لمنظمة التجارة العالمية في جنيف .
[]:	يقع المقر الرئيسي في لمنظمة التجارة العالمية [جنيف] .
XML:	يقع المقر الرئيسي ل <a> WTO في جنيف .

Table 3: Example errors and correctly projected markers with GMT. In #1, a necessary Chinese verb “报道 (report)” is lost in the XML-marked translation, while tags (<e>, /e>) are also mismatched due to the word reordering of “记者 (reporter)”. In #2, []-marked translation fails to preserve the square brackets ([]) around the Arabic translation of “WTO” (marked by underline).

en → **Lang.**	Hu et al.	GMT - **TyDiQA F$_1$**		
		XML	[]	“ ”
Arabic (*ar*)	68.8	68.4	**71.7**	66.5
Bengali (*bn*)	58.6	64.8	64.1	**69.3**
Finnish (*fi*)	69.4	69.6	**70.8**	68.0
Indonesian (*id*)	75.5	76.0	**78.6**	77.3
Korean (*ko*)	56.8	55.6	59.0	**59.6**
Russian (*ru*)	49.5	65.7	**66.1**	52.3
Swahili (*sw*)	69.1	**70.4**	70.1	70.1
Telugu (*te*)	**70.2**	69.0	67.3	67.9
AVG	64.7	67.4	**68.5**	66.4

Table 4: Comparison of different markers on TyDiQA-GoldP by training on the translated *projected data only*. Overall, square brackets ([]) have the best transfer learning performance.

Impact on Transfer Learning. We next evaluate the impact of different marker choices on the performance of cross-lingual transfer. The results on the TyDiQA dataset are presented in Table 4. On average, square brackets ([]) have the best transfer learning performance. We also directly compare with the projection data released by Hu et al., which utilizes XML tags and a Google internal translation system in the year 2020 to translate QA datasets. More results on NER and event extraction tasks, and comparison with the alignment-based projection methods are presented in Table 8.

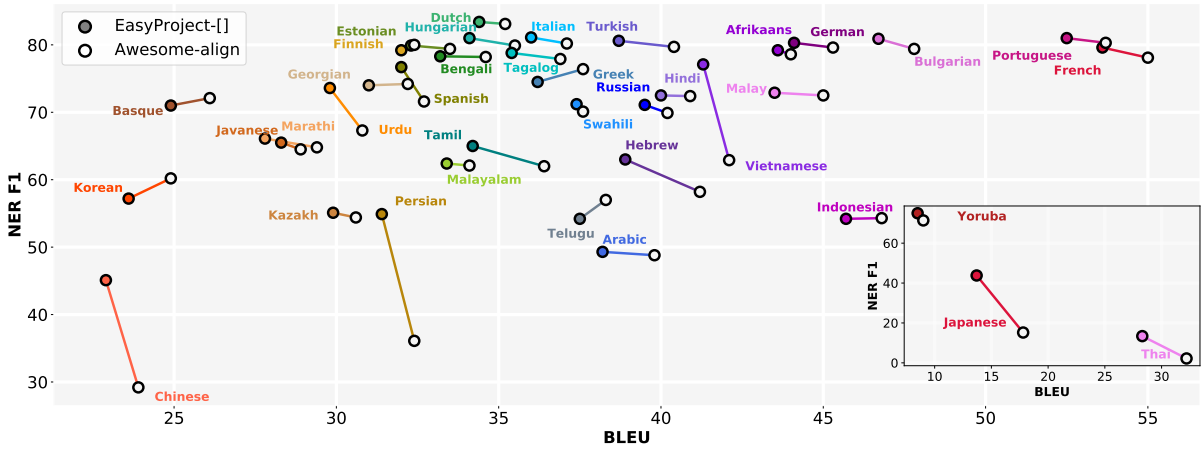

Figure 2: Comparison of translation quality (x-axis) and end-task performance (y-axis) for different label projection methods on the WikiANN dataset using NLLB translation system. EasyProject ● (§4) outperforms alignment-based approach ○ on F_1 scores for most languages, though inserting markers degrades translation quality. The experimental setting is detailed in §5.1.

4 EASYPROJECT

Based on our analysis, we develop an optimized version of the mark-then-translate method, which we call EASYPROJECT.[5] Our improvements target the two weaknesses of the marker-based approach: (1) special markers may get lost during translation; and (2) although square brackets ([]) show strong performance, they don't carry the correspondence between original spans and the ones in the translation (e.g., [Churchill] was born in [England] in [1874].), as the XML tags (e.g., <a> Churchill was born in England in <c> 1874 </c>.). If multiple annotated spans with different labels exist in one sentence, it is challenging to assign labels to the projected entities in the translation, as word order can change between languages.

4.1 Fine-tuning NLLB

To improve the robustness of the MT system in handling special markers, we further fine-tuned the NLLB model on synthetic data. We utilize parameter-efficient fine-tuning by only updating the last layer of the encoder and decoder, which take 4.2% of all parameters. We found fine-tuning 200 steps is sufficient to improve the projection rate on TyDiQA dataset from 70% to 96.4% while maintaining the translation quality.

Creating Synthetic Data. We first construct a parallel corpus where the special markers are inserted around the corresponding name entities in source and target sentences, with following steps.

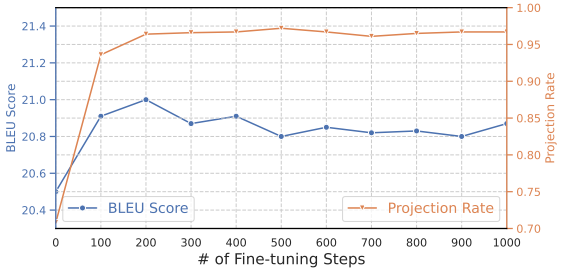

Figure 3: The changes of projection rate and translation quality (measured by BLEU score) as fine-tuning more steps. We found 200 steps are sufficient to boost the projection rate while maintaining translation quality.

1. Detect named entities on the English side of the parallel corpus, using the SpaCy NER system,[6] which covers 18 types of NER labels.
2. Translate the English entity names into the target language, and use string matching to find the corresponding entities from the target sentence. Given a pair of entities in the source and target sentences, we add square brackets ([]) around both of them.
3. Select all sentence pairs that contain more than one []-marked entity. We also sort the rest of the data based on length, and include the top-k sentence pairs. In total, we use 5,000 sentence pairs for each language pair.

We utilize the training data of NLLB model[7] as the source of the parallel sentences, and use the sentence pairs from high-resource language pairs (*en* to {*de,es,nl,zh,ar*}), which are selected based on the CoNLL-2002/2003 ({*de,es,nl*}) and ACE

[5] This name was inspired by Daumé III (2007).

[6] https://spacy.io/models/en#en_core_web_sm

[7] https://huggingface.co/datasets/allenai/nllb

datasets ({*zh,ar*}).

Parameter-efficient Fine-tuning. To save compute and preserve the translation ability of the model, we only update the weights in the last layer of the encoder and decoder for 200 steps, with a learning rate of 5e-5 and a batch size of 8, which takes around 2 minutes on an A40 GPU. The changes in projection rate and the translation quality during the fine-tuning process are shown in Figure 3. The fine-tuned NLLB model is able to improve the projection rate on TyDiQA from 70% to 96.4%. TyDiQA is particularly challenging due to its relatively long sentences, and the translation model sometimes may ignore the inserted markers. By fine-tuning on high-resource languages, we found the model is able to generalize well to the other language pairs. In pilot study, we notice that fine-tuning on low-resource languages, such as African languages in MasakhaNER corpus, doesn't generalize well and leads to lower translation quality. We will release all the fine-tuned models.

Fuzzy String Matching. To identify the corresponding labels when more than one projected entity exists in the translation, we design a fuzzy string-matching method. We first translate each annotated span in the original sentence independently, resulting in a set of labeled mentions. To identify labels for the unlabeled spans in the `[]`-marked translation, we compare each unlabeled span with the labeled mentions using the `ratio()` function in the `difflib` library. [8] Two strings are considered matched if they have >50% matched subsequences, and the associated label is assigned to the bracketed span. We also experiment with matching span labels left-to-right based on their relative position in the text. The results are shown in Table 5. Using fuzzy string-matching leads to overall better performance since it can assign the span labels more accurately.

Putting all the improvements together, we call this improved variant of marker-based method **EASYPROJECT** for **easi**ly **project**ing labels.

5 Experiments

In this section, we comprehensively evaluate the effectiveness of EASYPROJECT and analyze the key factors that impact the performance of cross-lingual transfer learning.

[8] https://docs.python.org/3/library/difflib.html#difflib.SequenceMatcher.ratio

	Original NLLB		Fine-tuned NLLB	
	Proj. Rate	F_1	Proj. Rate	F_1
`[]`+Fuzzy String Match	87.0%	62.6	93.7%	**62.9**
`[]`+Match by Sequence	87.7%	62.6	**94.4%**	62.3

Table 5: A comparison of varied methods to translate sentences and assign labels on the devset of MasakhaNER 2.0 corpus. "Proj.Rate" denotes the projection rate, which is defined in §3.2.

5.1 Comparison to Alignment-based Method

We first compare EasyProject with the traditional pipeline approach based on state-of-the-art bilingual word alignment models. For both methods, we apply a simple filtering rule that removes sentences with different numbers of annotations before and after projection.

Bilingual Word Alignment. We experiment with two state-of-the-art neural word aligners: (1) the unsupervised version of **Awesome-align** (Dou and Neubig, 2021) and its supervised version, which we extended from 5 to 39 languages for this paper, and (2) **QA-align** (Nagata et al., 2020) which formulates the word alignment problem as SQuAD-style QA task. More details on the word alignment models can be found in Appendix C.

Transfer Learning Results. As summarized in Figure 2, EasyProject outperforms alignment-based projection for most languages, even though

Fine-tune$_{en}$		NLLB+Aligner	NLLB+Markers	
Ref.	mDeBERTa	Awesome-align	XML	EasyProject.
56.9	55.0	63.2 (+8.2)	63.8 (+8.8)	**64.3** (+9.3)

Table 6: Average results over 18 African languages on the MasaKhaNER 2.0 corpus, as two languages are not supported by NLLB model. The mDeBERTa model is used here sinces it has the strongest performance on African languages in the original paper (Adelani et al., 2022), where the "Ref" column is from. Full results on each languages are in Table 11 in Appendix B .

Model	Ref.	Fine-tune$_{en}$	GMT+EasyProject
mT5$_{large}$	58.2	61.2	**68.5** (+7.3)
mT5$_{XL}$	65.1	62.9	**68.6** (+5.7)
XLM-R$_{large}$	63.3	64.3	**68.9** (+4.6)

Table 7: Average results for mT5 models over 39 target languages on WikiANN. The XLM-R model is listed for comparison. "Ref." column is the performance from prior work (He et al.; Xue et al.).

en → Lang.		Fine-tune$_{en}$		NLLB+Word Align			NLLB+Markers		GMT+Word Align			GMT+Markers	
		Ref.	XLM$_R$	QAali.	Awes.	Awes$_{ft}$	XML	EProj. (Δ_{XLM_R})	QAali.	Awes.	Awes$_{ft}$	XML	EProj. (Δ_{XLM_R})
NER	*yo*	41.3	37.1	-	73.2	78.0	68.7	77.7 (+40.6)	-	72.1	66.1	71.8	73.8 (+36.7)
	ja	18.3	18.0	19.3	23.4	22.4	17.3	45.5 (+27.5)	19.3	23.0	22.6	42.0	43.5 (+25.5)
	zh	25.8	27.1	47.6	36.0	34.0	46.2	46.6 (+19.5)	45.2	43.3	39.6	43.8	45.9 (+18.8)
	th	1.5	0.7	-	2.6	2.5	8.8	14.0 (+13.3)	-	1.2	1.3	14.7	15.1 (+14.4)
	ur	54.2	63.6	-	71.6	71.8	74.4	74.7 (+11.1)	-	70.2	72.3	76.3	74.7 (+11.1)
	he	54.1	56.0	-	58.3	58.1	61.1	63.4 (+7.4)	-	59.6	60.2	63.7	67.1 (+11.1)
	ms	69.8	64.1	-	69.4	72.7	74.6	73.9 (+9.8)	-	73.0	73.8	73.2	74.1 (+10.0)
	my	51.3	53.5	-	61.6	62.9	56.2	60.1 (+6.6)	-	60.2	60.1	57.0	62.0 (+8.5)
	ar	43.7	48.5	49.3	48.7	47.6	45.9	50.5 (+2.0)	50.7	50.9	51.2	51.3	56.3 (+7.8)
	jv	58.4	62.3	-	64.8	61.6	67.7	67.0 (+4.7)	-	64.6	68.8	69.2	69.8 (+7.5)
	tl	72.2	73.0	-	80.1	78.8	79.5	79.3 (+6.3)	-	80.4	80.4	79.9	80.0 (+7.0)
	hi	71.0	69.5	-	73.9	73.4	73.8	74.4 (+4.9)	-	75.6	76.0	75.9	75.7 (+6.2)
	ka	68.9	68.8	-	74.5	75.0	70.4	74.2 (+5.4)	-	73.5	73.2	72.7	74.7 (+5.9)
	bn	76.3	75.1	-	80.5	80.2	80.1	80.7 (+5.6)	-	82.0	81.7	80.6	80.9 (+5.8)
	ta	56.9	58.8	-	63.1	63.7	53.8	63.5 (+4.7)	-	62.4	63.2	63.9	64.3 (+5.5)
	eu	62.1	63.6	-	70.3	70.0	64.7	68.7 (+5.1)	-	69.8	66.5	67.5	69.0 (+5.4)
	ko	58.0	57.9	-	61.1	60.6	59.4	58.0 (+0.1)	-	62.9	62.4	61.7	61.9 (+4.0)
	mr	64.1	63.9	-	63.6	64.0	62.9	64.9 (+1.0)	-	62.6	61.2	64.0	67.1 (+3.2)
	sw	70.0	68.5	-	70.6	71.5	70.1	69.7 (+1.2)	-	70.2	71.5	72.2	70.7 (+2.2)
	vi	77.2	74.2	-	70.4	65.8	77.8	77.5 (+3.3)	-	70.4	67.2	77.5	76.0 (+1.8)
	te	52.3	55.6	-	57.7	56.3	51.8	55.9 (+0.3)	-	57.4	56.8	57.6	57.4 (+1.8)
	id	52.3	52.4	-	53.5	55.3	52.7	53.1 (+0.7)	-	52.7	55.0	57.3	53.9 (+1.5)
	ml	65.8	63.5	-	63.2	64.8	56.5	61.3 (-2.2)	-	61.9	63.0	68.1	64.3 (+0.8)
	es	68.8	74.8	-	72.2	70.2	73.3	71.7 (-3.1)	-	71.3	72.6	73.5	75.6 (+0.8)
	de	77.9	79.4	79.7	79.5	79.6	81.5	80.0 (+0.6)	79.5	80.0	79.4	79.8	80.2 (+0.8)
	kk	49.8	53.5	-	53.5	53.9	40.4	54.0 (+0.5)	-	53.2	55.1	51.3	54.2 (+0.7)
	fr	79.0	80.1	80.7	79.8	80.9	80.9	81.5 (+1.4)	79.6	80.7	79.4	81.5	80.8 (+0.7)
	af	77.6	78.6	-	79.3	78.4	79.1	79.4 (+0.8)	-	79.1	78.9	79.0	79.2 (+0.6)
	et	78.0	79.6	-	80.7	79.2	80.2	79.9 (+0.3)	-	80.2	79.6	78.6	80.1 (+0.5)
	hu	79.3	81.0	-	80.3	79.8	79.7	80.4 (-0.6)	-	79.9	79.7	80.6	80.7 (-0.3)
	fi	78.6	80.6	-	81.0	80.9	80.4	79.8 (-0.8)	-	80.7	79.7	78.8	80.3 (-0.3)
	it	81.1	81.3	-	80.5	80.5	81.9	81.2 (-0.1)	-	80.3	80.4	81.1	80.9 (-0.4)
	tr	78.9	80.3	-	80.6	81.0	80.1	79.5 (-0.8)	-	80.1	80.2	81.5	79.6 (-0.7)
	nl	84.3	84.1	-	83.4	83.3	83.0	83.4 (-0.7)	-	83.5	82.9	83.0	83.1 (-1.0)
	bg	81.2	82.1	-	80.2	78.8	81.9	82.5 (+0.4)	-	80.9	79.7	82.5	80.6 (-1.5)
	pt	79.6	82.0	-	80.9	80.4	82.6	81.9 (-0.1)	-	79.0	80.2	80.6	80.1 (-1.9)
	ru	71.5	71.1	-	68.9	68.1	70.0	70.3 (-0.8)	-	67.4	66.8	67.4	68.2 (-2.9)
	el	77.2	79.3	-	76.3	75.7	77.7	74.1 (-5.2)	-	73.1	75.2	76.2	75.0 (-4.3)
	fa	61.1	64.3	-	41.5	47.3	51.3	52.1 (-12.2)	-	52.9	52.4	45.5	52.0 (-12.3)
	AVG	63.3	64.3	-	66.4	66.4	66.1	68.4 (+4.1)	-	66.7	66.6	68.3	68.9 (+4.6)
QA	*ko*	31.9	56.1	-	36.9	36.4	64.8	67.7 (+11.6)	-	37.6	37.1	60.9	65.0 (+8.9)
	bn	64.0	66.0	-	71.1	72.6	63.7	69.6 (+3.6)	-	73.6	69.3	74.4	71.0 (+5.0)
	fi	70.5	69.7	-	74.9	74.0	73.0	73.3 (+3.6)	-	74.9	74.9	73.1	74.0 (+4.3)
	te	70.1	72.9	-	74.9	74.6	69.9	78.3 (+5.4)	-	75.9	69.9	77.0	77.0 (+4.1)
	ar	67.6	72.4	74.2	76.8	76.4	72.7	75.9 (+3.5)	74.0	76.3	76.6	75.8	76.4 (+4.0)
	sw	66.1	69.9	-	73.0	74.7	72.4	73.4 (+3.5)	-	72.3	73.4	73.6	73.5 (+3.6)
	ru	67.0	66.5	-	70.9	71.5	69.1	70.4 (+3.9)	-	71.6	69.7	70.2	69.8 (+3.3)
	id	77.4	78.0	-	81.6	81.1	79.6	80.3 (+2.3)	-	80.4	81.3	78.9	79.7 (+1.7)
	AVG	64.3	68.9	-	70.0	70.2	70.7	73.6 (+4.7)	-	70.3	69.0	73.0	73.3 (+4.4)

Event Extraction		Fine-tune$_{en}$	NLLB+Word Align			NLLB+Markers		GMT+Word Align			GMT+Markers	
		XLM$_R$	QAali.	Awes.	Awes$_{ft}$	XML	EProj. (Δ_{XLM_R})	QAali.	Awes.	Awes$_{ft}$	XML	EProj. (Δ_{XLM_R})
Arabic	*Entity*	69.2	74.1	74.2	74.2	73.6	73.8 (+4.6)	74.4	74.3	74.0	73.7	74.0 (+4.8)
	Relation	28.1	34.7	35.2	30.8	30.8	30.7 (+2.6)	34.8	33.1	34.2	31.8	33.7 (+5.6)
	Trig-I	42.7	43.5	43.0	44.7	43.3	43.7 (+1.0)	43.6	44.2	43.7	43.8	44.0 (+1.3)
	Trig-C	40.0	41.4	41.3	42.9	41.1	41.8 (+1.8)	41.8	42.6	42.0	41.5	42.0 (+2.0)
	Arg-I	33.5	37.1	38.1	37.6	37.1	37.6 (+4.1)	37.7	37.9	37.6	36.9	37.8 (+4.3)
	Arg-C	30.8	34.3	35.4	34.7	34.9	34.8 (+4.0)	34.6	34.5	34.5	34.1	35.2 (+4.4)
	AVG	40.7	44.2	44.5	44.1	43.5	43.7 (+3.0)	44.5	44.4	44.3	43.6	44.5 (+3.8)
Chinese	*Entity*	59.1	67.8	70.7	70.7	73.5	73.5 (+14.4)	67.1	68.8	70.6	70.2	71.0 (+11.9)
	Relation	20.4	31.2	34.7	35.9	37.3	37.8 (+17.4)	30.7	28.2	30.1	35.6	28.4 (+8.0)
	Trig-I	25.0	48.6	55.3	56.2	49.3	52.5 (+27.5)	43.7	53.5	50.0	50.7	52.6 (+27.6)
	Trig-C	23.9	45.6	52.1	52.0	46.1	49.0 (+25.1)	40.8	50.0	46.6	47.4	49.3 (+25.4)
	Arg-I	28.6	42.6	42.8	40.9	43.6	42.3 (+13.7)	38.7	39.6	39.4	39.8	40.1 (+11.5)
	Arg-C	28.1	40.3	41.2	39.4	42.1	40.8 (+12.7)	37.3	38.4	38.2	38.2	38.2 (+10.1)
	AVG	30.9	46.0	49.5	49.2	48.7	49.3 (+18.4)	43.1	46.4	45.8	47.0	46.6 (+15.7)

Table 8: Cross-lingual transfer experiments from English to target languages on three tasks: (1) NER on WikiAnn, (2) QA on TyDiQA-GoldP, and (3) Event extraction on ACE. Overall, EasyProject (EProj.) achieves stronger performance compared to the alignment-based methods, and also outperforms using XML tags. "Fine-tune$_{en}$" refer to the zero-shot baselines, where the models are trained only on English data. "Ref" column is the performance from prior work: QA in Hu et al. (2020) and NER from He et al. (2021b); "XLM$_R$" is our reimplementation using XLM-RoBERTa$_{large}$, which Δ is calculated against. We show the results that use Google Translation (GMT) and NLLB model separately. For '-', the language is not supported by the supervised word aligner. Cells are colored by Δ: -10 -5 -1 0 +1 +5 +10.

English:	He was buried in Woodlawn Cemetery in Bronx , New York City .
Alignment-based:	他被埋葬在纽约市 布朗 克斯 的伍德劳恩公墓。
EasyProject:	他被埋葬在纽约市 布朗 克斯 的伍德劳恩公墓。

Table 9: In this example, the correct projection should be "纽约市 布朗 克斯". The outputs from two label projection methods are marked. For the alignment-based projection, "克斯" is incorrectly missed. The translations are based on Google Translation.

span markers degrade translation quality. In Table 6 and 8, we show that EasyProject almost always outperforms alignment-based projection on NER, QA, and the more challenging event extraction tasks, when training on a combination of English data and the translated projected data in target languages. In addition, we find that EasyProject generally performs better than using XML tags, as the former has less impact on the translation quality. We also notice the relatively low zero-shot performance in *ja*, *zh*, and *th* on WikiAnn dataset, which is consistent with scores reported in prior literature (He et al., 2021b). We suspect this is due to their distinct script systems, and adding EasyProject data brings significant improvements to all of them – Japanese (+25.5 F_1), Chinese (+18.8 F_1), and Thai (+14.4 F_1). EasyProject (GMT) also improves the performance of $mT5_{large}$ and $mT5_{XL}$ by 7.3 and 5.7 F_1 on average across all target languages, and $mT5_{XXL}$ by 2.2 F_1 on a subset of 8 languages, as shown in Table 7. Full results of the mT5 model are provided in Table 22 in Appendix.

Accuracy of Projected Annotations. To answer why EasyProject can outperform alignment-based method even though it degrades translation quality, we manually inspect 400 sentences sampled from the WikiANN training set. EasyProject correctly projects 100% and 97.5% of the label spans, when using Google Translation and NLLB, respectively. Whereas the traditional method based on Awesome-align only achieves 97.5% and 93.4% accuracy. We found EasyProject can more accurately preserve the boundaries of the label span. For the alignment-based method, most errors are caused by partial or missed alignments, as demonstrated in Table 9. More analyses are provided in the Appendix G.2.

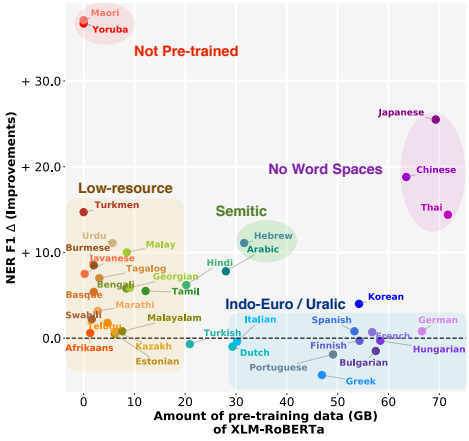

Figure 4: NER ΔF_1 (EasyProject+GMT over $Fine\text{-}tune_{en}$) vs. amount of pre-training data (GB) for $XLM\text{-}RoBERTa_{large}$.

5.2 Size of Pre-training Data vs. Improvement in Performance

Figure 4 shows improvements in NER F_1 using EasyProject vs. size of data for each language in XLM-RoBERTa's pre-training corpus. EasyProject provides larger improvements on low-resource languages and languages without whitespaces. For high-resource languages in the Indo-European (e.g., Germanic and Romance) or Uralic families, using projected data struggles to significantly improve over a strong fine-tuning baseline.

5.3 Transfer from non-English Languages

Recent work has suggested that English may not always be the best language to transfer from (Turc et al., 2021). We demonstrate that the marker-based method is not limited to English-centric transfer learning; rather, it can be used for transfer learning from any language to any language provided with the availability of multilingual MT systems. In Figure 5, we show the relative F_1 improvements of using EasyProject over fine-tuning on source language only for 9 different languages (81 directions in total), leveraging the multilingual capabilities of NLLB. Fine-tuning models only on source-language data does not work well when transferring to or from Chinese, consistent with observations from Hu et al. (2020). The marker-based method addresses this problem by providing substantial improvements in F_1 on the WikiANN dataset for Chinese. Transferring to Arabic and from Russian are also challenging, but again, the marker-based method greatly boosts performance.

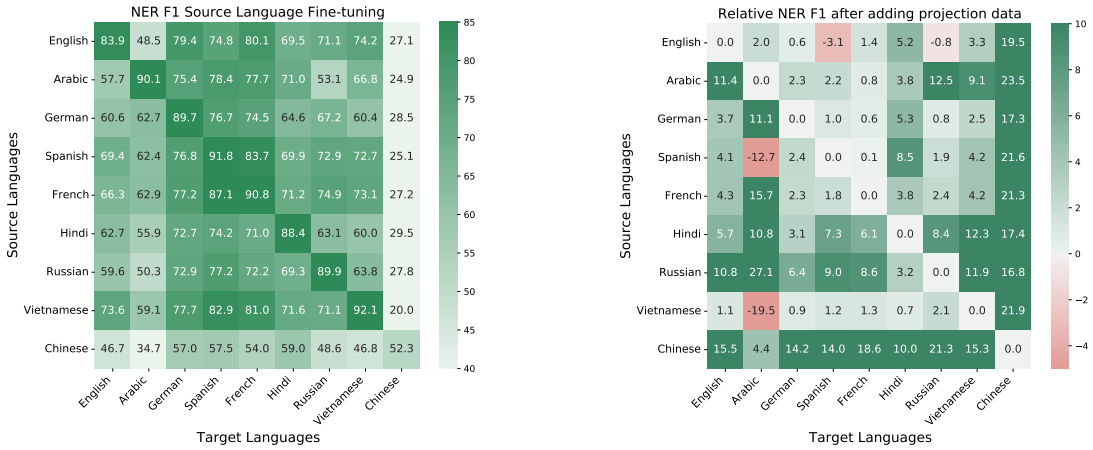

Figure 5: (a) NER F_1 for fine-tuning on different source and target languages. (b) The relative difference of F_1 for models trained on *source and projected data together* over *source data only*, when using EasyProject with the NLLB translation system. EasyProject can be used for transfer learning from any language to any language provided with the availability of multilingual MT systems.

5.4 More Experiments and Analyses in the Appendix

We also compare EasyProject against MulDA (Liu et al., 2021) (§F.1) and bitext projection (F.2), as well as evaluating it on low-resource languages: Māori and Turkmen (§F.3). In addition, we analyze the two branches of label projection methods from other aspects, including projection rate (§G.2) and translation speed (§G.3). Due to space limits, we present all of them in the appendix. On the data side, we fix a sentence splitting issue for 12 extremely long sentences in the ACE05 Arabic testset (§E). This issue is noticed by other researchers (Huang et al., 2022b) as well. We will release the improved ACE Arabic dataset to the community.

6 Conclusion

In this paper, we presnet a thorough empirical assessment of various approaches for cross-lingual label projection. We also design an improved variant of the mark-and-translate method, which we call EASYPROJECT. Experiments on 57 target languages and three well-studied NLP tasks show that EasyProject consistently outperforms the alignment-based methods and effectively improves the performance of cross-lingual transfer.

Limitations

While our study shows that EasyProject can effectively translate the source sentences with special markers inserted to the target languages, using the Google Translation and NLLB model, it is unclear whether all translation models can work well when special markers are inserted. To generalize this approach to future MT systems, we design a simple and computationally efficient approach to improve the robustness of MT systems in handling special markers. However, the translation quality for the marker-inserted text still falls behind the original text. We leave the work of further optimizing the translation quality as future work.

Acknowledgements

This material is based upon work supported by the NSF (IIS-2052498) and IARPA via the BETTER and HIATUS programs (2019-19051600004, 2022-22072200004). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF, 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.

References

David Ifeoluwa Adelani, Graham Neubig, Sebastian Ruder, Shruti Rijhwani, Michael Beukman, Chester Palen-Michel, Constantine Lignos, Jesujoba O Alabi, Shamsuddeen H Muhammad, Peter Nabende, et al. 2022. MasakhaNER 2.0: Africa-centric transfer learning for named entity recognition. In *Pro-*

ceedings of the 2022 Conference on Empirical Methods in Natural Language Processing.

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. Gate: graph attention transformer encoder for cross-lingual relation and event extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015a. 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*.

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015b. 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*.

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*.

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*.

Mihaela Bornea, Lin Pan, Sara Rosenthal, Radu Florian, and Avirup Sil. 2021. Multilingual transfer learning for QA using translation as data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yang Chen and Alan Ritter. 2021. Model selection for cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*.

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

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of 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*.

Jacob Devlin. 2018. Multilingual BERT readme document.

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*.

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*.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Ramy Eskander, Smaranda Muresan, and Michael Collins. 2020. Unsupervised cross-lingual part-of-speech tagging for truly low-resource scenarios. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond English-centric multilingual machine translation.

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*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. DeBERTaV3: Improving deberta using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *ArXiv preprint*.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021b. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

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. Proceedings of Machine Learning Research.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022a. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022b. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

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*.

David Kamholz, Jonathan Pool, and Susan Colowick. 2014. PanLex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*.

Phillip Keung, Yichao Lu, Julian Salazar, and Vikas Bhardwaj. 2020. Don't use English dev: On the zero-shot cross-lingual evaluation of contextual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

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*.

Wuwei Lan, Chao Jiang, and Wei Xu. 2021. Neural semi-Markov CRF for monolingual word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Kyungjae Lee, Kyoungho Yoon, Sunghyun Park, and Seung-won Hwang. 2018. Semi-supervised training data generation for multilingual question answering. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

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*.

Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq Joty, Luo Si, and Chunyan Miao. 2021. MulDA: A multilingual data augmentation framework for low-resource cross-lingual NER. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Mehrad Moradshahi, Giovanni Campagna, Sina Semnani, Silei Xu, and Monica Lam. 2020. Localizing open-ontology QA semantic parsers in a day using machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Masaaki Nagata, Katsuki Chousa, and Masaaki Nishino. 2020. A supervised word alignment method based on cross-language span prediction using multilingual BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Jian Ni, Georgiana Dinu, and Radu Florian. 2017. Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Massimo Nicosia, Zhongdi Qu, and Yasemin Altun. 2021. Translate & Fill: Improving zero-shot multilingual semantic parsing with synthetic data. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Tong Niu, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. OneAligner: Zero-shot cross-lingual transfer with one rich-resource language pair for low-resource sentence retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *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*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. ZeRo: Memory optimizations toward training trillion parameter models. In *International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE.

Parker Riley, Isaac Caswell, Markus Freitag, and David Grangier. 2020. Translationese as a language in "multilingual" NMT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. 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*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

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*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of English in zero-shot cross-lingual transfer. *ArXiv preprint*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. LDC2006T06.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint*.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

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*.

Mahsa Yarmohammadi, Shijie Wu, Marc Marone, Haoran Xu, Seth Ebner, Guanghui Qin, Yunmo Chen, Jialiang Guo, Craig Harman, Kenton Murray, Aaron Steven White, Mark Dredze, and Benjamin Van Durme. 2021. Everything is all it takes: A multipronged strategy for zero-shot cross-lingual information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

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*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Ran Zhou, Xin Li, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. Conner: Consistency training for cross-lingual named entity recognition. *EMNLP*.

Imed Zitouni, Jeffrey Sorensen, Xiaoqiang Luo, and Radu Florian. 2005. The impact of morphological stemming on Arabic mention detection and coreference resolution. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*.

A Data statistics for CoNLL-2002/2003

The statistics of the CoNLL-2002/2003 multilingual NER dataset are provided in Table 10.

	CoNLL 2002/2003
# of Lang.	3
# of Docs	–
# of Sent.	14k/3.2k/3.4k
Avg. Length	–/14.5
Avg. # of Spans	1.7

Table 10: The detailed statistics of train/dev/test sets for CoNLL-2002/2003 dataset. **Avg. Length** represents the average number of tokens in each article/sentence, and **Avg. # of Spans** denotes the average number of annotated spans in each sentence.

B Full Results on MasakhaNER2.0

MasahkaNER2.0 is a NER dataset in the news domain, including the annotations on 20 African languages. Following the setting in the original paper (Adelani et al., 2022), we use CoNLL-03 dataset (Tjong Kim Sang and De Meulder, 2003) as the source corpus, and train the mDeBERTv3 (He et al., 2021a) model on it. Then the trained model is evaluated on the test set of MasahkaNER2.0, with a focus on the PER, ORG, and LOC types.

Language	Ref.	Fine-tune$_{en}$	+Awes.	+XML	+EasyProj.
Bambara(bam)	38.4	37.1	45.0	44.3	45.8
Ghomala(bbj)	45.8	43.3	–	–	–
Ewe(ewe)	76.4	75.3	78.3	77.8	78.5
Fon(fon)	50.6	49.6	59.3	60.2	61.4
Hausa(hau)	72.4	71.7	72.7	71.6	72.2
Igbo(ibo)	61.4	59.3	63.5	59.6	65.6
Kinyarwanda(kin)	67.4	66.4	63.2	70.8	71.0
Luganda(lug)	76.5	75.3	77.7	77.9	76.7
Luo(luo)	53.4	35.8	46.5	50.0	50.2
Mossi(mos)	45.4	45.0	52.2	53.6	53.1
Chichewa(nya)	80.1	79.5	75.1	73.5	75.3
Naija(pcm)	75.5	75.2	–	–	–
chiShona(sna)	37.1	35.2	69.5	56.3	55.9
Kiswahili(swa)	87.9	87.7	82.4	81.7	83.6
Setswana(tsn)	65.8	64.8	73.8	72.9	74.0
Akan/Twi(twi)	49.5	50.1	62.7	64.7	65.3
Wolof(wol)	44.8	44.2	54.5	58.9	58.9
isiXhosa(xho)	24.5	24.0	61.7	71.9	71.1
Yoruba(yor)	40.4	36.0	38.1	36.8	36.8
isiZulu(zul)	44.7	43.9	68.9	74.8	73.0
Averaged Perf.	56.9	55.0	63.2	63.8	64.3
Averaged Proj. Rate	-	-	86.9%	77.5%	93.7%

Table 11: F1 scores on MasakhaNER2.0 using NLLB translation model. We skip Ghomala and Naija as they are not supported by NLLB.

C Details of Word Alignment Models

Awesome-align. This aligner (Dou and Neubig, 2021), when used in the unsupervised setting, primarily relies on the normalized similarity scores of all word pairs between the two sentences, which are calculated based on pre-trained multilingual word embeddings taken from specific Transformer layers. In the supervised setting, with access to parallel text, Awesome-align can be further improved by fine-tuning towards a set of self-training and language model objectives. We include experiments of both the unsupervised (*Awesome*) and supervised (*Awesome$_{ft}$*) versions of Awesome-align based on multilingual BERT, which has shown to achieve better word alignment results than XLM-RoBERTa$_{\text{base}}$. For the supervised version, we fine-tune an individual Awesome-align model for each of the 39 target languages in WikiANN using parallel sentences sampled from the M2M model's (Fan et al., 2021) training datasets: CCAligned (El-Kishky et al., 2020) and CCMatrix (Schwenk et al., 2021b). Specifically, we randomly sample 200k parallel sentences from the CCAligned corpus for language pairs from English to {*te*, *ka*, *kk*, *my*, *th*, *yo*}, and the rest from the CCMatrix.

We use the codebase[9] from Dou and Neubig (2021) with the default softmax configuration to extract alignment. We do not apply the consistency optimization objective when fine-tuning the models because it may trade precision for recall, as suggested in the official instruction written by the authors.

QA-align. This is a state-of-the-art supervised approach (Nagata et al., 2020) that formulates the word alignment problem as a SQuAD-style question answering task by fine-tuning multilingual BERT. Specifically, given a word in the source sentence, the model predicts the aligned span in the target sentence and reconciles both source-to-target and target-to-source directions by averaging and thresholding probabilities. We trained the QA-align model for English to Arabic, German, French, Chinese, and Japanese, where gold annotated word alignment data is available.

We use the codebase from Nagata et al. (2020).[10] For the training data of word alignment between

[9] https://github.com/neulab/awesome-align

[10] https://github.com/nttcslab-nlp/word_align

Lang.	Train	Test
en-ar	40,288	9,280
en-de	300	208
en-fr	300	147
en-ja	653	357
en-zh	4,879	610

Table 12: Number of sentences in the train/dev sets of the annotated word alignment datasets.

en and {*de*, *zh*, *ja*, *fr*}, we use the same data as in Nagata et al. (2020). For *en-ar*, we use the GALE English-Arabic word alignment data from LDC[11], and use 80% of the sentence pairs for training. The data statistics can be found in Table 12.

D Implementation Details of IE models

We follow the same learning rates and number of epochs reported in prior work: Hu et al. (2020) for QA, He et al. (2021b) and Pfeiffer et al. (2020) for NER (the latter for *mi* and *tk*) and Yarmohammadi et al. (2021) for ACE. For WikiANN NER (Pan et al., 2017), CoNLL-2002/2003 NER (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003), MasakhaNER2.0 (Adelani et al., 2022), and TyDiQA-GoldP (Clark et al., 2020), we use the codebase from the XTREME benchmark (Hu et al., 2020),[12] and MasakhaNER2.0 [13], which is based on the Huggingface transformers library (Wolf et al., 2019). The hyperparameters of mDeBERTaV3 (276M) for MasakhaNER2.0 and XLM-RoBERTa$_{\text{large}}$(550M) and for other datasets are presented in Table 13 following (Hu et al., 2020; He et al., 2021b; Liu et al., 2021; Adelani et al., 2022). We report the average result of three random seeds and select models based on the English development set.

mT5 NER Model. Training mT5$_{\text{XXL}}$ (Xue et al., 2021) models, which have over 11 billion parameters, for the NER task is computationally challenging. We formulate the WikiANN NER task as generating a sequence of tokens with special entity tags (e.g. `<per>`, `</per>`) inserted around the entity spans. To fit the model into GPU memory

[11]LDC2014T05, LDC2014T10, LDC2014T14, LDC2014T19 LDC2013T10, LDC2013T14, LDC2014T03, LDC2014T08

[12]https://github.com/google-research/xtreme

[13]https://github.com/masakhane-io/masakhane-ner

	WikiANN	CoNLL	Masakha	TyDiQA-GoldP
Task	NER	NER	NER	QA
Epochs	5	10	5	3
Batch size	32	32	32	8
Learning rate	2e-5	2e-5	5e-5	3e-5
Warmup steps	0	0	0	500
Weight decay	0	0	0	0.0001

Table 13: Hyperparameters for fine-tuning the NER and QA models.

for training, we freeze the embedding layer and the bottom 12 layers of both encoder and decoder during fine-tuning. We also use the `DeepSpeed` (Rajbhandari et al., 2020) ZeRo3 with 32-bits configurations. We first fine-tune the model on English data for 20 epochs with a learning rate of 1e-4. To speed up the training process, we initialize the model from the English fine-tuned checkpoint and further fine-tune it on the combination of English and EasyProject + GMT data with a learning rate of 5e-5 for 5 epochs. We report results of mT5$_{\text{large}}$ by averaging over three random seeds. We use one random seed for the XL and XXL models due to the heavy computing cost. Experiment results of average performance across languages are shown in Table 7, and results of each language are reported in Table 22 in Appendix.

ACE05. For ACE05 event extraction (Walker et al., 2006), we use the OneIE joint model v0.4.8 codebase[14] with the same hyperparameters as Yarmohammadi et al. (2021). For evaluation, we use the OneIE scoring tool to report F_1 scores for entities, relations, event triggers identification (Trig-I) and classification (Trig-C), argument role identification (Arg-I) and classification (Arg-C). We train models on the combination of English and projected Chinese data from scratch in the Chinese experiment and select the model based on the English development set. In the Arabic experiment, we initialize the model from the English fine-tuned checkpoint. We fine-tune the argument role classifier for event extraction tasks (Entity, Trig-I, Trig-C, Arg-I, Arg-C) and relation classifier in relation task for 5 epochs. We set the learning rate of task-specific classifiers at 1e-6 and the encoder at 5e-4. During the decoding process of relation classification, we only consider the joint model's relation and entity prediction scores.

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

	en	*en*†	*zh*	*ar*	*ar*†
Sent.	19,216	19,216	547	321	321
Entity	47,554	28,996	2,388	2,897	1,971
Relation	7,159	4,925	672	469	411
Trig.	4,419	3,125	190	232	232
Arg.	6,607	5,128	332	447	348
Tok/Sent	14.2	14.2	37.4	32.4	32.4

Table 14: Statistic of the ACE05 English(*en*) training set and Chinese(*zh*)/Arabic(*ar*) test set. We hire an native Arabic speaker to fix the Arabic test set by sentence-splitting the 12 articles that miss punctuation. †: remove pronoun entities and related annotations in the events and relations.

On the Arabic data annotation side, the ACE Arabic data contains language-specific annotations on pronoun entities due to morphological stemming (Zitouni et al., 2005), where we observe individual Arabic letter (prefix or suffix) is annotated as a pronoun entity. Because such annotations don't exist in English data, the label projection process may cause inconsistency in translated-Arabic data. Thus, we remove the pronoun entities in both Arabic test data and English training data for the Arabic experiment. The complete statistics of the Arabic test set is in Table 14. We report the average results of three random seeds.

E Fixing Issues in the Arabic ACE Data

The ACE data are pre-processed using the code from Lin et al. (2020). We use the same document splits as Lin et al. (2020) for English (ACE05-E$^+$) and Chinese (ACE05-CN). For Arabic, we use the document splits from Lan et al. (2020) following Yarmohammadi et al. (2021).

In the processed Arabic test set from Yarmohammadi et al. (2021), we observed 12 extremely long sentences with an average length of 381 tokens, which are significantly longer than the rest of the sentences with an average length of 28. This issue was also independently noticed by Huang et al. (2022b). A closer look reveals that these 12 sentences are 12 full articles in the original LDC release, which appear to be missing punctuation. We hire a native Arabic speaker to manually split them into sentences, resulting in 106 additional sentences. The data statistics are shown in Table 14. Because the ACE data is licensed, we will release the processing script instead.

F More Experiments

In this section, we present more experiments to compare EasyProject with other approaches.

F.1 Comparison to MulDA

Table 15 shows a direct comparison of EasyProject with MulDA (Liu et al., 2021), another translation-based label projection approach that has been recently proposed for NER. MulDA replaces named entities with placeholders one by one, such as '`PER0`' and '`LOC1`', then invokes the MT system separately for each entity to translate and project the data. Thus, MulDA is more time-consuming and costly than the EasyProject, which only requires one invocation of the MT system per sentence. We find that EasyProject outperforms MulDA in German, Spanish, and Dutch at much less time cost. In this experiment, we follow MulDA's experimental setup, which uses the CoNLL NER dataset and trains only on the projected data.

In terms of translation speed, we calculate the relative time cost of EasyProject compared to translating the original sentences in CoNLL English data using the NLLB model on one A40 GPU. In Table 15, we observe marker-based (XML and `[]`) translation takes 1.2× and 1.3× longer of time to translate, due to the additional markers in both the input and output. More analysis of the translation speed is provided in Appendix G.3.

Method	MT	*de*	*es*	*nl*	*time*
MulDA	GMT	73.9	75.5	79.6	-
MulDA	NLLB	74.5	73.5	77.5	2.4×
XML	GMT	74.3	**77.1**	79.8	-
XML	NLLB	**75.3**	74.9	78.3	1.2×
EasyProject	GMT	74.9	70.3	**79.9**	-
EasyProject	NLLB	75.2	73.0	77.5	1.3×

Table 15: Comparison of MulDA (Liu et al., 2021) and EasyProject on CoNLL NER (F_1), using *projected data only*. "*time*": relative time cost compared to translating the original sentence.

F.2 Comparison to Bitext Projection

Besides translation-projection, another alternative is bitext-projection, in which bilingual parallel corpora are used in place of a machine translation system. For example, we can apply a trained English IE model to the English side of the bilingual parallel corpus, then use word alignment to project the

automatically predicted labels to the corresponding sentences in target languages.

In Table 16, we show that bitext-projection improves F_1 of WikiANN NER on 6 out of 8 languages used in (Yarmohammadi et al., 2021) over the fine-tuning baseline (Fint-tune$_{en}$), but is outperformed by EasyProject. For this experiment, we randomly sample 100,000 parallel sentences for each of the eight languages from WikiMatrix (Schwenk et al., 2021a), an automatically mined bitext corpus from Wikipedia that matches the domain of WikiANN. We use an XLM-RoBERTa$_{\text{large}}$ NER model trained on WikiANN English data with 83.9 F_1 to generate named entity labels, and then apply Awesome-align to project labels to the target language. Finally, we train the XLM-RoBERTa$_{\text{large}}$ model on English and bitext-projected data together for 2 epochs (Bitext$_{100k}$).

Bitext$_{100k}$ loses 13.9 F_1 score for Vietnamese (*vi*), most likely due to Awesome-align projection errors being magnified by fine-tuning on 100,000 projected sentences. One surprising finding is that the Bitext$_{100k}$ improves by an absolute 4.4 F_1 score on Spanish and 1.1 F_1 on Russian. Translation-projection approaches struggle on these two languages as shown in Table 8.

Lang.	FT$_{en}$	EasyProject	Bitext$_{100k}$	EasyProject +Bitext$_{100k}$
ar	48.5	56.3 (+7.6)	52.6 (+4.1)	51.3 (+2.8)
de	79.4	80.2 (+0.8)	81.0 (+1.6)	81.4 (+2.0)
es	74.8	75.6 (+0.8)	79.2 (+4.4)	77.7 (+2.9)
fr	80.1	80.8 (+0.7)	80.5 (+0.4)	82.4 (+2.3)
hi	69.5	75.7 (+6.2)	68.6 (-0.9)	74.6 (+5.1)
ru	71.1	68.2 (-2.9)	72.2 (+1.1)	68.6 (-2.5)
vi	74.2	76.0 (+1.8)	60.3 (-13.9)	77.5 (+3.3)
zh	27.1	45.9 (+18.8)	31.7 (+4.6)	44.5 (+17.4)
AVG	65.6	69.8 (+4.2)	65.8 (+0.2)	69.8 (+4.2)

Table 16: Comparison of NER F_1 on WikiANN between Bitext-Projection with 100,000 bilingual sentence pairs and EasyProject with GMT.

F.3 Experiments on Low-resource Languages

To investigate the effectiveness of label projection on very low-resource languages (Pfeiffer et al., 2020), we conduct experiments on Māori (*mi*) and Turkmen (*tk*), which are not covered by the pre-trained language models (i.e., XLM-RoBERTa and mBERT) and have a small number of Wikipedia articles (∼1.2k for Māori and ∼0.5k for Turkman). As shown in Table 17, EasyProject improves F_1 score by an absolute 22.7 F_1 score on Māori and 5.9 F_1 on Turkmen compared to fine-tuning on English data only. We also include a lexicon-based baseline, replacing English words with their word-to-word translations based on PanLex (Kamholz et al., 2014), a commonly used multilingual dictionary. Both EasyProject and Awesome-align significantly outperform the word-level translations, likely because word-level translations still follow the English word orders and fail to capture the variation of word orders in Māori and Turkmen. For example, Māori has a verb-subject-object word order, while Turkmen uses a subject-object-verb. The improvement is less significant on Turkmen than Māori, potentially because Turkmen is close to Turkish, which is covered by both mBERT and XLM-RoBERTa. This is also a plausible reason why Awesome-align that uses mBERT did better on Turkmen.

Method	MT	Māori(*mi*)	Turkmen(*tk*)
mBERT†	-	21.8	47.2
XLM-RoBERTa$^{\dagger}_{\text{base}}$	-	15.9	43.4
XLM-RoBERTa$_{\text{large}}$	-	30.3	52.2
+ word-translation	PanLex	42.5	53.8
+ Awesome-align	GMT	46.1	**60.7**
+ EasyProject	GMT	**53.0**	58.1

Table 17: F_1 scores for cross-lingual NER from English to two very **low-resource languages** on WikiANN. PanLex is a bilingual dictionary. †: English fine-tuning results reported in (Pfeiffer et al., 2020).

G More Analysis on EasyProject

Here, we present more analysis of the EasyProject method in comparison with the traditional pipeline approach based on word alignment.

G.1 Translation Quality

To further measure the impact of adding special markers on the translation quality for the NLLB model, we adopt the evaluation setup used by NLLB (Costa-jussà et al., 2022) which utilizes the professional human-translated FLORES-200 parallel corpora (1000 sentences per language). For the marker-based approaches ("XML" and "[]"), special markers are removed from the outputs before calculating the BLEU scores. Table 18 presents the BLEU scores for the original NLLB model (3.3B) and the NLLB model further fine-tuned with three

Language	NLLB			$\text{NLLB}_{\text{finetune}}$			Language	NLLB			$\text{NLLB}_{\text{finetune}}$		
	Orig	XML	[]	Orig	XML	[]		Orig	XML	[]	Orig	XML	[]
Afrikaans(af)	44.0	44.3	43.6	45.9	45.4	45.5	Luo(luo)	15.6	15.6	15.3	16.5	16.0	15.9
Arabic(ar)	39.8	38.5	38.2	39.0	36.7	37.6	Malayalam(ml)	34.1	33.2	33.4	39.1	36.6	37.8
Bulgarian(bg)	47.8	47.5	46.7	48.4	45.0	46.7	Mossi(mos)	6.4	6.0	6.4	6.5	6.3	6.4
Bambara(bm)	10.5	10.5	10.3	10.4	10.0	10.2	Marathi(mr)	29.4	27.9	27.8	29.8	27.1	28.1
Bengali(bn)	34.6	32.6	33.2	35.3	31.1	33.0	Malay(ms)	45.0	43.9	43.5	45.1	43.6	43.5
German(de)	45.3	44.2	44.1	45.7	43.3	44.6	Burmese(my)	16.2	16.1	14.0	20.4	16.0	17.7
Ewe(ee)	16.3	16.0	16.4	16.7	15.9	16.2	Dutch(nl)	35.2	34.9	34.4	35.0	33.0	33.8
Greek(el)	37.6	36.7	36.2	37.1	34.6	35.5	Chichewa(ny)	17.4	17.6	17.0	20.5	19.5	19.9
Spanish(es)	32.7	32.3	32.0	32.1	31.0	31.3	Portuguese(pt)	53.7	53.0	52.5	54.6	52.2	53.1
Estonian(et)	33.5	32.9	32.3	33.0	30.5	31.3	Russian(ru)	40.2	38.8	39.5	40.1	36.9	38.9
Basque(eu)	26.1	26.3	24.9	30.8	28.8	29.2	Kinyarwanda(rw)	24.6	24.2	22.7	26.9	24.6	25.7
Persian(fa)	32.4	31.8	31.4	32.3	30.3	30.8	Shona(sn)	19.4	18.2	18.2	19.6	16.6	17.5
Finnish(fi)	32.4	31.9	32.0	32.9	29.5	31.2	Swahili(sw)	37.6	37.1	37.4	40.1	38.2	39.1
Benin(fon)	5.3	5.0	5.7	5.3	4.0	5.6	Tamil(ta)	36.4	34.6	34.2	37.7	34.6	36.3
French(fr)	55.0	54.7	53.6	55.3	52.9	53.6	Telugu(te)	38.3	37.3	37.5	39.1	37.4	38.2
Hausa(ha)	29.2	28.6	28.0	29.4	28.0	28.5	Thai(th)	32.2	30.7	28.3	32.9	28.8	29.9
Hebrew(he)	41.2	39.6	38.9	39.4	35.6	37.1	Tagalog(tl)	36.9	35.1	35.4	34.8	32.8	33.4
Hindi(hi)	40.9	39.1	40.0	41.3	38.3	40.6	Tswana(tn)	25.8	25.8	25.2	26.5	24.5	26.5
Hungarian(hu)	35.5	34.7	34.1	35.5	33.0	33.8	Turkish(tr)	40.4	39.2	38.7	41.0	37.7	38.9
Indonesian(id)	46.8	45.7	45.7	46.5	43.9	45.3	Twi(tw)	16.1	16.1	15.8	16.7	16.5	16.2
Igbo(ig)	19.7	19.8	19.3	20.2	19.5	20.0	Urdu(ur)	30.8	29.5	29.8	30.6	29.0	29.8
Italian(it)	37.1	36.2	36.0	36.0	33.9	34.6	Vietnamese(vi)	42.1	41.5	41.3	41.9	39.8	40.8
Japanese(ja)	17.8	17.0	13.7	19.9	18.0	18.8	Wolof(wo)	9.2	9.2	9.4	9.1	9.7	9.2
Javanese(jv)	28.9	28.1	28.3	29.4	27.9	28.7	Xhosa(xh)	24.2	23.8	22.9	26.5	23.8	24.9
Georgian(ka)	32.2	31.2	31.0	32.6	28.7	31.3	Yoruba(yo)	9.0	10.4	8.5	6.8	6.4	8.0
Kazakh(kk)	30.6	30.1	29.9	33.2	30.1	31.4	Chinese(zh)	23.9	24.2	22.9	28.0	26.2	26.8
Korean(ko)	24.9	23.9	23.6	25.2	22.2	24.4	Zulu(zu)	30.3	29.1	29.2	30.8	27.8	28.5
Ganda(lg)	12.5	13.0	12.4	13.0	13.2	13.1							
Average	30.2	29.5	29.1	30.9	28.8	29.7							

Table 18: BLEU score of NLLB on FLORES-200 (Costa-jussà et al., 2022) dev set (1000 sentences per language). We compare three types of translations: original translation (Orig), inserted with XML and [] special markers. We also fine-tuned NLLB with different markers using the method described in Appendix 4.1. We found that the fine-tuned NLLB model using square brackets has the least negative impact on translation quality

types of parallel sentences (original, inserted with XML and [] markers).

As there is no gold NER annotation on the parallel corpus, we first train a NER model based on $\text{XLM-RoBERT}_{\text{large}}$ on the WikiAnn dataset, achieving an F_1 score of 83.9. We then apply the trained model to the English side of the parallel corpus and apply the EasyProject method to translate sentences into the target language. After removing the special markers from the translation outputs, we use the `sacreBLEU`[15] to calculate the BLEU scores by comparing the translations against gold references. We follow the NLLB evaluation setting and use multilingual tokenizations (flores200).[16]

[15] https://github.com/mjpost/sacrebleu

[16] https://github.com/mjpost/sacrebleu/blob/master/CHANGELOG.md

G.2 Projection Rate

We then compare the projection rate for all label projection methods, for which we divide the number of annotations after projection by the number of annotations occurring in the original training data. We also include the average number of successfully projected sentences after filtering out the incorrect ones, which have a different number of annotations compared to the source sentence. For example, the source sentence has a `LOC` and a `PER` entity, but the projected sentence has two `LOC` entities. Such sentences will be filtered. For QA-align in WikiANN NER, we show the average statistics for 5 languages {*ar, de, fr, ja, zh*} that have supervised training data.

As shown in Table 21, Google Translation (GMT) is very robust in handling special markers, and EasyProject has a nearly perfect 100% projec-

English #1:	Dean of Wolverhampton (1373 - 1394)
Alignment-based:	沃尔弗 汉普顿 伯爵, 1373 - 1394 年
EasyProject:	沃尔弗 汉普顿 伯爵 1373 - 1394
English #2:	Pino Daniele (1955 - 2015)
Alignment-based:	皮诺· 丹尼埃尔 (Pino Daniele , 1955 - 2015 年)
EasyProject:	皮诺· 丹尼尔 (1955 - 2015 年)

Table 19: Examples from WikiANN dataset using NLLB translation. The outputs from two projection methods and correct answers are marked. In #1, the alignment-based method incorrectly misses the "沃尔弗", which is a part of the translation for "Wolverhampton". In #2, for the alignment-based method, the Chinese translation ("皮诺· 丹尼埃尔") and the original English span ("Pino Daniele") occur together in the translation. Alignment-based method incorrectly misses the correct projection "皮诺· 丹尼埃尔" and project to "Pino Daniele".

	#Inputs	#Outputs	Time (sec)
Original	279,678	335,963	4,452
XML	460,002	468,815	5,486
[]	326,294	379,796	4,107
Entity	64,293	72,309	1,553

Table 20: Number of tokens in the three types of input sentences: original CoNLL NER English training data, adding XML and [] special markers; and their corresponding translations in German. Time is the total translation clockwise time in seconds.

tion rate, higher than any word alignment-based method. Our manual inspection of 100 sentences, randomly sampled from the WikiANN training set for English to Chinese projection, also reveals that GMT+EasyProject successfully projects all the sentences without mistakes on any target named entities, whereas Awesome-align only projected 94 sentences and caused 4 entity projection errors. According to our manual analysis, EasyProject is less likely to introduce errors than the word alignment-based method because the use of special markers encourages full-span projection.

We found that most errors are caused by partial or missed alignments, which often occur when a span contains multiple words, a sentence contains many spans, or when both a Chinese transliteration and the original English name occur together in the translated sentence, which is a correct way to translate but poses challenges for label projection. More examples of alignment errors can be found in Table 19 in Appendix.

G.3 Translation Speed

Additional special markers added to the source sentence will affect the translation speed. In Table 20, we show the number of tokens in the input and translation output. We use the CoNLL-2002/2003 NER English training set as the source sentences, and translate them into German. All sentences are tokenized by the NLLB tokenizer. We also show the translation time per sentence and per entity on an A40 GPU with a batch size of 32.

We estimate using XML tags takes 1.2× time compared to translating the original sentences, and EasyProject takes 1.3× time as it requires the additional translation of each entity span, for identifying the label correspondence.

		NLLB+Word Aligner			NLLB+Markers		GMT+Word Aligner			GMT+Markers	
		QAalign	Awesome.	Awesome$_{ft}$	XML	EProj.	QAalign	Awesome.	Awesome$_{ft}$	XML	EProj.
NER	# Sents	18,486(5)	18,274	18,587	13,959	19,470	19,187(5)	19,003	19,408	20,000	20,000
	Proj. Rate	92.4(5)	91.4	92.9	69.8	97.4	96.0(5)	94.8	98.2	100	100
Event	# Sents	15,491	14,840	15,857	7,308	16,846	16,264	16,631	16,903	19,185	19,185
	Proj. Rate	80.6	77.2	82.5	38.0	87.7	90.4	92.6	93.6	99.9	99.9
QA	# Sents	-	3,613	3,654	1,573	3,564	-	3,623	3,649	3,695	3,695
	Proj. Rate	-	97.8	98.9	42.6	96.4	-	97.8	99.1	100	100

Table 21: Diagnosis analysis of projected data based on two metrics: number of sentences and the percentage of the projected annotations (Proj. Rate). For QA-align in NER, we show 5 languages {*ar,de,fr,ja,zh*}.

Lang.	XLM-R$_{large}$	+EasyProject	mT5$_{large}$	+EasyProject	mT5$_{XL}$	+EasyProject	mT5$_{XXL}$	+EasyProject
af	78.6	79.2 (+0.6)	79.2	81.0 (+1.8)	77.2	79.6 (+2.4)	-	-
ar	48.5	56.3 (+7.8)	53.1	66.1 (+13.0)	57.4	68.0 (+10.6)	62.2	66.1 (+3.9)
bg	82.1	80.6 (-1.5)	58.5	77.0 (+18.5)	61.5	76.1 (+14.6)	-	-
bn	75.1	80.9 (+5.8)	57.3	76.2 (+18.9)	65.7	75.8 (+10.1)	-	-
de	79.4	80.2 (+0.8)	75.6	77.9 (+2.3)	75.9	77.6 (+1.7)	76.5	77.3 (+0.8)
el	79.3	75.0 (-4.3)	61.6	81.6 (+20.0)	79.4	77.2 (-2.2)	-	-
es	74.8	75.6 (+0.8)	85.7	87.0 (+1.2)	86.3	85.3 (-1.0)	85.6	86.4 (+0.8)
et	79.6	80.1 (+0.5)	71.8	72.8 (+1.0)	71.7	73.2 (+1.4)	-	-
eu	63.6	69.0 (+5.4)	64.0	68.0 (+4.0)	64.0	74.1 (+10.1)	-	-
fa	64.3	52.0 (-12.3)	47.0	67.5 (+20.5)	46.1	64.9 (+18.8)	-	-
fi	80.6	80.3 (-0.3)	74.6	78.0 (+3.4)	73.5	79.2 (+5.7)	-	-
fr	80.1	80.8 (+0.7)	84.6	84.9 (+0.3)	83.8	84.2 (+0.4)	83.4	84.2 (+0.8)
he	56.0	67.1 (+11.1)	53.3	63.3 (+10.1)	57.9	66.1 (+8.2)	-	-
hi	69.5	75.7 (+6.2)	70.1	76.0 (+5.9)	74.8	77.1 (+2.2)	76.0	76.4 (+0.4)
hu	81.0	80.7 (-0.3)	76.0	82.0 (+6.0)	76.5	80.0 (+3.5)	-	-
id	52.4	53.9 (+1.5)	77.6	77.9 (+0.3)	82.2	82.3 (+0.1)	-	-
it	81.3	80.9 (-0.4)	86.2	86.4 (+0.1)	86.4	85.5 (-1.0)	-	-
ja	18.0	43.5 (+25.5)	28.3	38.3 (+10.0)	29.8	38.0 (+8.3)	-	-
jv	62.3	69.8 (+7.5)	72.4	75.7 (+3.2)	72.9	72.3 (-0.6)	-	-
ka	68.8	74.7 (+5.9)	60.6	72.2 (+11.6)	67.1	72.5 (+5.4)	-	-
kk	53.5	54.2 (+0.7)	32.7	53.1 (+20.4)	26.1	51.7 (+25.5)	-	-
ko	57.9	61.9 (+4.0)	33.7	39.1 (+5.4)	30.6	44.7 (+14.1)	-	-
ml	63.5	64.3 (+0.8)	42.1	65.1 (+23.0)	42.5	63.9 (+21.3)	-	-
mr	63.9	67.1 (+3.2)	49.6	57.4 (+7.9)	53.9	55.6 (+1.8)	-	-
ms	64.1	74.1 (+10.0)	79.3	79.6 (+0.3)	80.5	79.4 (-1.1)	-	-
my	53.5	62.0 (+8.5)	35.0	38.7 (+3.7)	31.9	33.0 (+1.1)	-	-
nl	84.1	83.1 (-1.0)	84.2	85.5 (+1.3)	83.5	84.1 (+0.5)	-	-
pt	82.0	80.1 (-1.9)	83.0	82.9 (+0.0)	83.5	82.7 (-0.8)	-	-
ru	71.1	68.2 (-2.9)	55.3	70.8 (+15.6)	59.8	70.1 (+10.3)	65.6	72.8 (+7.2)
sw	68.5	70.7 (+2.2)	65.9	66.4 (+0.5)	66.8	73.9 (+7.1)	-	-
ta	58.8	64.3 (+5.5)	49.5	61.7 (+12.1)	52.6	63.3 (+10.7)	-	-
te	55.6	57.4 (+1.8)	47.4	57.5 (+10.1)	51.3	57.8 (+6.5)	-	-
th	0.7	15.1 (+14.4)	2.0	3.8 (+1.8)	2.0	7.4 (+5.4)	-	-
tl	73.0	80.0 (+7.0)	80.6	81.6 (+1.0)	81.9	83.2 (+1.3)	-	-
tr	80.3	79.6 (-0.7)	68.8	69.7 (+1.0)	71.4	68.8 (-2.6)	-	-
ur	63.6	74.7 (+11.1)	51.4	65.4 (+14.0)	56.9	67.0 (+10.1)	-	-
vi	74.2	76.0 (+1.8)	81.4	83.0 (+1.6)	81.7	82.0 (+0.4)	82.4	79.6 (-2.8)
yo	37.1	73.8 (+36.7)	75.7	82.3 (+6.6)	75.5	78.4 (+3.0)	-	-
zh	27.1	45.9 (+18.8)	31.1	39.7 (+8.7)	31.6	39.8 (+8.2)	36.3	43.2 (+6.9)
AVG	64.3	68.9 (+4.6)	61.2	68.5 (+7.4)	62.9	68.6 (+5.7)	71.0	73.3 (+2.3)

Table 22: Cross-lingual NER F_1 on WikiANN for mT5 and XLM-RoBERTa$_{large}$. Due to the computing limit, we run the largest mT5$_{XXL}$ model on 8 languages which were chosen following Yarmohammadi et al. (2021). The performance is averaged over 3 runs for XLM-R$_{large}$ and mT5$_{large}$ models, and 1 run for mT5$_{XL}$ and mT5$_{XXL}$ models. Models are fine-tuned on a combination of English and EasyProject data with Google Translation.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitation section

☑ A2. Did you discuss any potential risks of your work?
Limitation section

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Appendix A-F

☑ B1. Did you cite the creators of artifacts you used?
Appendix A-F

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix F

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix E

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
we use existing published datasets.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Table 14

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Table 1 & 10

C ☑ Did you run computational experiments?

Section 3 & 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix E and H.3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix E

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix E

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix C and Appendix H.1

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Appendix F

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Appendix F

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Appendix F