“Wikily” Supervised Neural Translation Tailored to Cross-Lingual Tasks

Mohammad Sadegh Rasooli1, Chris Callison-Burch2, Derry Tanti Wijaya3

1Microsoft
2Department of Computer and Information Science, University of Pennsylvania
3Department of Computer Science, Boston University
mrasooli@microsoft.com, ccb@seas.upenn.edu, wijaya@bu.edu

Abstract

We present a simple but effective approach for leveraging Wikipedia for neural machine translation as well as cross-lingual tasks of image captioning and dependency parsing without using any direct supervision from external parallel data or supervised models in the target language. We show that first sentences and titles of linked Wikipedia pages, as well as cross-lingual image captions, are strong signals for a seed parallel data to extract bilingual dictionaries and cross-lingual word embeddings for mining parallel text from Wikipedia. Our final model achieves high BLEU scores that are close to or sometimes higher than strong supervised baselines in low-resource languages; e.g. supervised BLEU of 4.0 versus 12.1 from our model in English-to-Kazakh. Moreover, we tailor our “wikily” supervised translation models to unsupervised image captioning, and cross-lingual dependency parser transfer. In image captioning, we train a multitasking machine translation and image captioning pipeline for Arabic and English from which the Arabic training data is a translated version of the English captioning data, using our wikily-supervised translation models. Our captioning results on Arabic are slightly better than that of its supervised model. In dependency parsing, we translate a large amount of monolingual text, and use it as artificial training data in an annotation projection framework. We show that our model outperforms recent work on cross-lingual transfer of dependency parsers.

1 Introduction

Developing machine translation models without using bilingual parallel text is an intriguing research problem with real applications: obtaining a large volume of parallel text for many languages is hard if not impossible. Moreover, translation models could be used in downstream cross-lingual tasks in which annotated data does not exist for some languages. There has recently been a great deal of interest in unsupervised neural machine translation (e.g. Artetxe et al. (2018a); Lample et al. (2018a,c); Conneau and Lample (2019); Song et al. (2019a); Kim et al. (2020); Tae et al. (2020)). Unsupervised neural machine translation models often perform nearly as well as supervised models when translating between similar languages, but they fail to perform well in low-resource or distant languages (Kim et al., 2020) or out-of-domain monolingual data (Marchisio et al., 2020). In practice, the highest need for unsupervised models is to expand beyond high resource, similar European language pairs.

There are two key goals in this paper: Our first goal is developing accurate translation models for low-resource distant languages without any supervision from a supervised model or gold-standard parallel data. Our second goal is to show that our machine translation models can be directly tailored to downstream natural language processing tasks. In this paper, we showcase our claim in cross-lingual image captioning and cross-lingual transfer of dependency parsers, but this idea is applicable to a wide variety of tasks.

We present a fast and accurate approach for learning translation models using Wikipedia. Unlike unsupervised machine translation that solely relies on raw monolingual data, we believe that we should not neglect the availability of incidental supervisions from online resources such as Wikipedia. Wikipedia contains articles in nearly 300 languages and more languages might be added in the future, including indigenous languages and dialects of different regions in the world. Different from similar recent work (Schwenk et al., 2019a), we do not rely on any supervision from supervised translation models. Instead, we leverage the fact that many first sentences in linked Wikipedia pages are rough translations, and furthermore, many captions of
A summary of our contribution is as follows: 1) We propose a simple, fast and effective approach towards using the Wikipedia monolingual data for machine translation without any explicit supervision. Our mining algorithm easily scales on large comparable data using limited computational resources. We achieve very high BLEU scores for distant languages, especially those in which current unsupervised methods perform very poorly. 2) We propose novel methods for leveraging our current translation models in image captioning. We show that how a combination of translating caption training data, and multi-task learning with English captioning as well as translation improves the performance. Our results on Arabic shows results slightly superior to that of a supervised captioning model trained on gold-standard datasets. 3) We propose a novel modification to the annotation projection method to be able to leverage our translation models. Our results on dependency parsing performs better than previous work in most cases, and performs similarly to using gold-standard parallel datasets.

Our translation and captioning code and models are publicly available online\(^1\).

2 Background

**Supervised neural machine translation**  Supervised machine translation uses a parallel text \( P = \{(s_i, t_i)\}_{i=1}^n \) in which each sentence \( s_i \in l_1 \) is a translation of \( t_i \in l_2 \). **Neural machine translation** uses sequence-to-sequence models with attention (Cho et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) for which the likelihood of training data is maximized by maximizing the log-likelihood of predicting each target word given its previous predicted words and source sequence:

\[
\mathcal{L}(P) = \sum_{i=1}^n \sum_{j=1}^{|t_i|} \log p(t_{i,j}|t_{i,k<j}, s_i; \theta)
\]

where \( \theta \) is a collection of parameters to be learned.

**Unsupervised neural machine translation**  Unsupervised neural machine translation does not have access to any parallel data. Instead, it tailors monolingual datasets \( M_{l_1} \) and \( M_{l_2} \) for learning multilingual language models. These language

models usually mask parts of every input sentence, and try to uncover the masked words (Devlin et al., 2019). The monolingual language models are used along with iterative back-translation (Hoang et al., 2018) to learn unsupervised translation. An input sentence \( s \) is translated to \( t' \) using current model \( \theta \), then the model assumes that \((t', s)\) is a gold-standard translation, and uses the same training objective as of supervised translation.

**Dependency parsing** Dependency parsing algorithms capture the best scoring dependency trees for sentences among an exponential number of possible dependency trees. A valid dependency tree for a sentence \( s = s_1, \ldots, s_n \) assigns heads \( h_i \) for each for word \( s_i \) where \( 1 \leq i \leq n \), \( 0 \leq h_i \leq n \) and \( h_i \neq i \). The zeroth word represents a dummy root token as an indicator for the root of the sentence. For more details about efficient parsing algorithms, we encourage the reader to see Kübler et al. (2009).

**Annotation projection** Annotation projection is an effective method for transferring supervised annotation from a rich-resource language to a low-resource language through translated text (Yarowsky et al., 2001). Having a parallel data \( \mathcal{P} = \{(s_i, t_i)\}_{i=1}^{n} \), and supervised source annotations for source sentences \( s_i \), we transfer those annotations through word translation links \( 0 \leq a_{i}^{(j)} \leq |t_i| \) for \( 1 \leq j \leq |s_i| \) where \( a_{i}^{(j)} = 0 \) shows a null alignment. The alignment links are learned in an unsupervised fashion using unsupervised word alignment algorithms (Och and Ney, 2003a). In dependency parsing, if \( h_i = j \) and \( a_{i}^{(j)} = k \) and \( a_{i}^{(k)} = m \), we project a dependency \( k \rightarrow m \) (i.e. \( h_m = k \)) to the target side. Previous work (Rasooli and Collins, 2017, 2019) has shown that annotation projection only works when a large amount of translation data exists. In the absence of parallel data, we create artificial parallel data using our translation models. Figure 2 shows an example of annotation projection using translated text.

### 3 Learning Translation from Wikipedia

The key component of our approach is to leverage the multilingual cues from linked Wikipedia pages across languages. Wikipedia is a great comparable data in which many of its pages explain entities in the world in different languages. In most cases, first sentences define or introduce the mentioned entity in that page (e.g. Figure 1). Therefore, we observe that many first sentence pairs in linked Wikipedia documents are rough translations of each other. Moreover, captions of images in different languages are usually similar but not necessarily direct translations of each other. We leverage this information to extract many parallel sentences from Wikipedia without using any external supervision. In this section, we describe our algorithm which is briefly shown in Figure 3.

#### 3.1 Data Definitions

For languages \( e \) and \( f \) in which \( e \) is English and \( f \) is a low-resource target language of interest, there are Wikipedia documents \( \{w_e = \{w_{1}^{(e)} \ldots w_{n}^{(e)}\}\} \) and \( \{w_f = \{w_{1}^{(f)} \ldots w_{m}^{(f)}\}\} \). We refer to \( w_{(i,j)}^{(l)} \) as the \( j \)th sentence in the \( i \)th document for language \( l \). A subset of these documents are aligned (using Wikipedia languages links). Thus we have an aligned set of document pairs in which we can easily extract many sentence pairs that are potentially translations of each other. A smaller subset \( \mathcal{F} \) is the set of first sentences in Wikipedia \( \{(w_{i}^{(e)}, w_{i}^{(f)})\} \) in which documents \( i \) and \( i' \) are linked and their first sentence lengths are in a similar range. In addition to text content, Wikipedia has a large set of images. Each image comes along with one or more captions, sometimes in different languages. A small subset of these images have captions both in English and the target language. We refer to this set as \( \mathcal{C} \). We use the set of all caption pairs (\( \mathcal{C} \)), title pairs (\( \mathcal{T} \)), and first sentences (\( \mathcal{F} \)) as the seed parallel data: \( \mathcal{S} = \mathcal{F} \cup \mathcal{C} \cup \mathcal{T} \).

#### 3.2 Bilingual Dictionary Extraction and Cross-Lingual Word Embeddings

Having the seed parallel data \( \mathcal{S} \), we run unsupervised word alignment (Dyer et al., 2013) in both English-to-target, and target-to-English directions. We use the intersected alignments to extract highly confident word-to-word connections. Finally, we pick the most frequently aligned word for each word in English as translation. This set serves as a bilingual dictionary \( \mathcal{D} \).

Given two monolingual trained word embeddings \( v_e \in \mathbb{R}^{N_e \times d} \) and \( v_f \in \mathbb{R}^{N_f \times d} \), and the extracted bilingual dictionary \( \mathcal{D} \), we use the method of Faruqui and Dyer (2014) to project these two embedding vectors to a shared cross-lingual space.\(^2\) This method uses a bilingual dictionary along with

\(^2\)There are more recent approaches such as (Lample et al., 2018b). Comparing different embedding methods is not the focus of this paper, thereby we leave further investigation to future work.
The International Crisis Group recently suggested moving responsibility for pension to state level, to eliminate some of the problems (Zeman et al., 2020). For more details about dependency parsing, we encourage the reader to see Kübler et al. (2009).

We use cross-lingual embedding vectors $v_i$ where $i \in \mathbb{R}^{V \times D}$ to a shared space $V$ using contextualized embeddings and attention. To alleviate this problem to some extent, we assume that a parallel data exists between language 1 and language 2, and we can use it both as an auxiliary and cross-lingual parallel data.

After filtering sentence pairs with different numerical values (e.g., sentences containing 2019 in the source and 1987 in the target), we use a modified version of cosine similarity between words:

$$
sim(s_i, t_j) = \begin{cases} 
1.0, & \text{if } (s_i, t_j) \in \mathcal{D} \\
\cos(s_i, t_j), & \text{otherwise}
\end{cases}
$$

Using the above definition of word similarity, we use the average-maximum similarity between pairs of sentences.

$$
score(s, t) = \frac{1}{n} \sum_{i=1}^{n} \max_{j=1}^{m} \sim(s_i, t_j)
$$

From a pool of candidates, we pick those pairs that have the highest score in both directions.

### 3.4 Leveraging Similar Languages

In many low-resource scenarios, the number of paired documents is very small, leading to a small number and often noisy extracted parallel sentences. To alleviate this problem to some extent, we assume to have another language $g$ in which $g$ has a large lexical overlap with the target language $f$ (such as $g=$Russian and $f=$Kazakh). We assume that a parallel data exists between language $g$ and English, and we can use it both as an auxiliary and cross-lingual parallel data.

After extracting extra lexical entries for the bilingual dictionaries: as shown in Figure 3, we supplement the extracted bilingual dictionary from seed parallel data with the bilingual dictionary extracted from related language parallel data.

### 3.5 Translation Model

We use a standard sequence-to-sequence transformer-based translation model (Vaswani et al., 2017) with a six-layer BERT-based (Devlin et al., 2019) encoder-decoder architecture.
We extract the final layer as a 7 × 7 vocabulary. All input and output token embeddings are summed up with the language id embedding. First tokens of every input and output sentence are shown by the language ID. Our training pipeline assumes that the encoder and decoder are shared across different languages, except that we use a separate output layer for each language in order to prevent input copying (Artetxe et al., 2018b; Sen et al., 2019). We pretrain the model on a tuple of three Wikipedia datasets for the three languages g, f, and e using the MASS model (Song et al., 2019a). The MASS model masks a contiguous span of input tokens, and recovers that span in the output sequence.

To facilitate multi-task learning with image captioning, our model has an image encoder that is used in cases of image captioning (more details in §4.1). In other words, the decoder is shared between the translation and captioning tasks. We use the pretrained ResNet-152 model (He et al., 2016) from Pytorch to encode every input image. We extract the final layer as a 7 × 7 grid vector (g ∈ R7×7×d), and project it to a new space by a linear transformation (g′ ∈ R49×d), and then add location embeddings (l ∈ R49×d) by using entry-wise addition. Afterwards, we assume that the 49 vectors are encoded text representations as if a sentence with 49 words occurs. This is similar to but not exactly the same as the Virtex model (Desai and Johnson, 2021).

3.6 Back-Translation: One-shot and Iterative

Finally, we use the back-translation technique to improve the quality of our models. Back-translation is done by translating a large amount of monolingual text to and from the target language. The translated texts serve as noisy input text along with the monolingual data as the silver-standard translations. Previous work (Sennrich et al., 2016b; Edunov et al., 2018) has shown that back-translation is a very simple but effective technique to improve the quality of translation models. Henceforth, we refer to this method as one-shot back-translation. Another approach is to use iterative back-translation (Hoang et al., 2018), the most popular approach in unsupervised translation (Artetxe et al., 2018b; Conneau and Lample, 2019; Song et al., 2019a). The main difference from one-shot translation is that the model uses an online approach, and updates its parameters in every batch.

We empirically find one-shot back-translation faster to train but with much less potential to reach a high translation accuracy. A simple and effective way to have both a reliable and accurate model is to first initialize a model with one-shot back-translation, and then apply iterative back-translation. The model that is initialized with a more accurate model reaches a higher accuracy.

4 Cross-Lingual Tasks

In this section, we describe our approaches for tailoring our translation models to cross-lingual tasks.

4.1 Cross-Lingual Image Captioning

Having gold-standard image captioning training data \( T = \{(I_i, c_i)\}_{i=1}^n \) where \( I_i \) is the image as pixel values, and \( c_i = c_i^{(1)}, \ldots, c_i^{(k_i)} \) as the textual description with \( k_i \) words, our goal is to learn a captioning model that is able to describe new (unseen) images. As described in §3.5, we use a transformer decoder from our translation model and a ResNet image encoder (He et al., 2016) for our image captioning pipeline. Unfortunately, annotated image captioning datasets do not exist in many languages. Having our translation model parameter \( \theta^* \), we can use its translation functionality to translate each caption \( c_i \) to \( c_i' = \text{translate}(c_i | \theta^*_c) \). Afterwards, we will have a translated annotated dataset \( T' = \{(I_i, c_i')\}_{i=1}^n \) in which the textual descriptions are not gold-standard but translations from the English captions. Figure 4 shows a real example from MS-Coco (Chen et al., 2015) in which Arabic translations are provided by our translation model. Furthermore, to augment our learning capability, we initialize our decoder with decoding parameters of \( \theta^*_c \), and also continue training with both English captioning and translation.

4.2 Cross-Lingual Dependency Parsing

Assuming that we have a large body of monolingual text, we translate that monolingual text to create artificial parallel data. We run unsupervised word alignments on the artificial parallel text. Following previous work (Rasooli and Collins, 2015; Ma and Xia, 2014), we run Giza++ (Och and Ney,
In this section, we provide details about our experimental settings and results for translation, captioning, and dependency parsing. We put more details about our settings as well as thorough analysis of our results in the supplementary material.

5 Experiments

In this section, we provide details about our experimental settings and results for translation, captioning, and dependency parsing. We put more details about our settings as well as thorough analysis of our results in the supplementary material.

5.1 Datasets and Settings

Languages We focus on four language pairs: Arabic-English, Gujarati-English, Kazakh-English, and Romanian-English. We choose these pairs to provide enough evidence that our model works in distant languages, morphologically-rich languages, as well as similar languages. As for similar languages, we use Persian for Arabic (written with very similar scripts and have many words in common), Hindi for Gujarati (similar languages), Russian for Kazakh (written with the same script), and Italian for Romanian (Romance languages).

Monolingual and Translation Datasets We use a shared SentencePiece vocabulary (Kudo and Richardson, 2018) with size 60K. Table 1 shows the sizes of Wikipedia data in different languages. For evaluation, we use the Arabic-English UN data (Ziemski et al., 2016), WMT 2019 data (Barault et al., 2019) for Gujarati-English and Kazakh-English, and WMT 2016 shared task data (Bojar et al., 2016) for Romanian-English. Following previous work (Sennrich et al., 2016a), diacritics are removed from the Romanian data. More details about other datasets and their sizes, we refer the reader to the supplementary material.

Pretraining We pretrain four models on 3-tuples of languages via a single NVIDIA GeForce RTX 2080 Ti with 11GB of memory. We create batches of 4K words, run pretraining for two million iterations where we alternate between language batches, and accumulate gradients for 8 steps. We use the apex library3 to use FP-16 tensors. This whole process takes four weeks in a single GPU. We use the Adam optimizer (Kingma and Ba, 2015) with inverse square root and learning rate of $10^{-4}$, 4000 warm-up steps, and dropout probability of 0.1.

Translation Training Table 1 shows the sizes of different types of datasets in our experiments. We pick comparable candidates for sentence pairs whose lengths are within a range of half to twice of each other. As we see, the final size of mined datasets heavily depends on the number of paired English-target language Wikipedia documents. We train our translation models initialized by pretrained models. More details about our hyperparameters are in the supplementary material. All of our evaluations are conducted using SacreBLEU (Post, 2018) except for en+ro in which we use BLEU score (Papineni et al., 2002) from Moses decoder scripts (Koehn et al., 2007) for the sake of comparison to previous work.

Image Captioning We use the Flickr (Hodosh et al., 2013) and MS-Coco (Chen et al., 2015) datasets for English4, and the gold-standard Arabic Flickr dataset (ElJundi et al., 2020) for evaluation. The Arabic test set has 1000 images with 3 captions

Table 1: Data sizes for different pairs. We use a sample of English sentences with similar sizes to each data.

<table>
<thead>
<tr>
<th>Direction</th>
<th>art-km</th>
<th>gu-km</th>
<th>kk-EN</th>
<th>ro-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreign docs</td>
<td>1.1M</td>
<td>1.1M</td>
<td>1.1M</td>
<td>2.1M</td>
</tr>
<tr>
<td>Paired docs</td>
<td>7.45k</td>
<td>7.1k</td>
<td>80k</td>
<td>270k</td>
</tr>
<tr>
<td>First sen.</td>
<td>205k</td>
<td>2.3k</td>
<td>52k</td>
<td>78k</td>
</tr>
<tr>
<td>Captions</td>
<td>92k</td>
<td>2.2k</td>
<td>1.9k</td>
<td>35k</td>
</tr>
<tr>
<td>Comparable pairs</td>
<td>0.1b</td>
<td>1.4M</td>
<td>32M</td>
<td>64M</td>
</tr>
<tr>
<td>Mined sen.</td>
<td>1.7m</td>
<td>49k</td>
<td>183k</td>
<td>675k</td>
</tr>
<tr>
<td>BT</td>
<td>2.1m</td>
<td>1.5m</td>
<td>2.2m</td>
<td>2.1m</td>
</tr>
<tr>
<td>Iterative BT</td>
<td>4.0m</td>
<td>3.8m</td>
<td>4.0m</td>
<td>6.1m</td>
</tr>
</tbody>
</table>

Figure 4: An image from MS-Coco (Chen et al., 2015) with gold-standard English captions, and Arabic translations from our wikity translation model.

---

3https://github.com/NVIDIA/apex

4We have also tried Conceptual Captions (Sharma et al., 2018) in our initial experiments but we have observed drops in performance. Previous work (Singh et al., 2020) have also observed a similar problem with Conceptual Captions as a noisy crawled caption dataset.
per image. We translate all the training datasets to Arabic for having translated caption data. The final training data contains 620K captions for about 125K unique images. Throughout experiments, we use the pretrained Resnet-152 models (He et al., 2016) from Pytorch (Paszke et al., 2019), and let it fine-tune during our training pipeline. Each training batch contains 20 images. We accumulate gradients for 16 steps, and use a dropout of 0.1 for the projected image output representations. Other training parameters are the same as our translation training. To make our pipeline fully unsupervised, we use translated development sets to pick the best checkpoint during training.

### Dependency Parsing
We use the Universal Dependencies v2.7 collection (Zeman et al., 2020) for Arabic, Kazakh, and Romanian. We use the Stanza (Qi et al., 2020) pretrained supervised models for getting supervised parse trees for Arabic and Romanian, and use the UDPipe (Straka et al., 2016) pretrained model for Kazakh. We translate about 2 million sentences from each language to English, and also 2 million English sentences to Arabic. We use a simple modification to Stanza to facilitate training on partially projected trees by masking dependency and label assignments for words with missing dependencies. All of our training on projected dependencies is blindly conducted with 100K training steps with default parameters of Stanza (Qi et al., 2020). As for gold-standard parallel data, we use our supervised translation training data for Romanian-English and Kazakh-English and use a sample of 2 million sentences from the UN Arabic-English data due to its large size that causes word alignment significant slowdown. For Kazakh wikily projections, due to low supervised POS accuracy, we use the projected POS tags for projected words and supervised tags for unprojected words. We observe a two percent increase in performance by using projected tags.

### 5.2 Translation Results
Table 2 shows the results of different settings in addition to baseline and state-of-the-art results. We see that Arabic as a clear exception needs more rounds of training: we train our Arabic model once again on mined data by initializing it by our back-translation model.\(^5\) We have not seen further improvement by back-translation. To have a fair comparison, we list the best supervised models for all language pairs (to the best of our knowledge). In low-resource settings, we outperform strong supervised models that are boosted by back-translation. In high-resource settings, our Arabic models achieve very high performance but regarding the fact that the parallel data for Arabic has 18M sentences, it is quite impossible to reach that level of accuracy.

Figure 5 shows a randomly chosen example from the Gujarati-English development data. As depicted, we see that the model after back-translation reaches to somewhat the core meaning of the sentence with a bit of divergence from exactly matching the reference. The final iterative back-translation output almost catches a correct translation. We also see that the use of the word “creative” is seen in Google Translate output, a model that is most likely trained on much larger parallel data than what is currently available for public use. In general, unsupervised translation performs very poorly compared to our approach in all directions.

### 5.3 Captioning Results
Table 4 shows the final results on the Arabic test set using the SacreBLEU measure (Post, 2018). First, we should note that similar to ElJundi. et al. (2020), we see lower scales of BLEU scores due to morphological richness in Arabic. We see that if we initialize our model with the translation model and multitask it with translation and also English captioning, we achieve much higher performance. It is interesting to observe that translating the English output on the test data to Arabic achieves a much lower result. This is a strong indicator of the strength of our approach. We also see that supervised translation fails to perform well. This might due to the UN translation training dataset which has a different domain from the caption dataset. Furthermore, we see that our model outperforms Google Translate which is a strong machine translation system, and that is actually what is being used as seed data for manual revision in the Arabic dataset. Finally, it is interesting to see that our model outperforms supervised captioning. Multi-tasking make translation performance slightly worse.

Figure 6 shows a randomly picked example with

---

\(^5\) We have seen that during multi-tasking with image captioning, the translation BLEU score for Arabic-English significantly improves. We initially thought that multi-tasking

Table 3: Dependency parsing results on the Universal Dependencies dataset (Zeman et al., 2020). Previous work has used different sub-versions of the Universal Dependencies data in which slight differences are expected.

Figure 5: An example of a Gujarati sentence and its outputs from different models, as well as Google Translate.

Figure 6: An example of different outputs in our captioning experiments both for English and Arabic, as well as Arabic translations of English outputs on the Arabic Flickr dataset (ElJundi. et al., 2020).

different model outputs. We see that the two outputs from our approach with multi-tasking are roughly the same but one of them as more syntactic order overlap with the reference while both orders are correct in Arabic as a free-word order language. The word برتقالية means “orange” which is close to حمراء that means “red”. The word means “slide” which is correct but other meanings of this word exist in the reference. In general, we observe...
Table 4: Image captioning results evaluated on the Arabic Flickr dataset (ElJundi et al., 2020) using SacreBLEU (Post, 2018). “pretrained” indicates initializing our captioning model with our translation parameters. that although superficially the BLEU scores for Arabic is low, it is mostly due to its lexical diversity, free-word order, and morphological complexity.

5.4 Dependency Parsing Results

Table 3 shows the results for dependency parsing experiments. We see that our model performs very high in Romanian with a UAS of 74 which is much higher than that of Ahmad et al. (2019) and slightly lower than that of Rasooli and Collins (2019) which uses a combination of multi-source annotation projection and direct model transfer. Our work on Arabic outperforms all previous work and performs even better than using gold-standard parallel data. One clear highlight is our result in Kazakh. As mentioned before, by projecting the part-of-speech tags, we achieve roughly 2 percent absolute improvement. Our final results on Kazakh are significantly higher than that of using gold-standard parallel data (7K sentences).

6 Related Work

Kim et al. (2020) has shown that unsupervised translation models often fail to provide good translation systems for distant languages. Our work solves this problem by leveraging the Wikipedia data. Using pivot languages has been used in previous work (Al-Shedivat and Parikh, 2019), as well as using related languages (Zoph et al., 2016; Nguyen and Chiang, 2017). Our work only explores a simple idea of adding one similar language pair. Most likely, adding more language pairs and using ideas from recent work might improve the performance.

Wikipedia is an interesting dataset for solving NLP problems including machine translation (Li et al., 2012; Patry and Langlais, 2011; Lin et al., 2011; Tušić et al., 2013; Barrón-Cedeño et al., 2015; Wijaya et al., 2017; Ruiter et al., 2019; Srinivasan et al., 2021). The WikiMatrix data (Schwenk et al., 2019a) is the most similar effort to ours in terms of using Wikipedia, but with using supervised translation models. Bitext mining has a longer history of research (Resnik, 1998; Resnik and Smith, 2003) in which most efforts are spent on using a seed supervised translation model (Guo et al., 2018; Schwenk et al., 2019b; Artetxe and Schwenk, 2019; Schwenk et al., 2019a; Jones and Wijaya, 2021). Recently, a number of papers have focused on unsupervised extraction of parallel data (Ruiter et al., 2019; Hangya and Fraser, 2019; Keung et al., 2020; Tran et al., 2020; Kuwanto et al., 2021). Ruiter et al. (2019) focus on using vector similarity of sentences to extract parallel text from Wikipedia. Their work does not leverage structural signals from Wikipedia.

Cross-lingual and unsupervised image captioning has been studied in previous work (Gu et al., 2018; Feng et al., 2019; Song et al., 2019b; Gu et al., 2019; Gao et al., 2020; Burns et al., 2020). Unlike previous work, we do not have a supervised translation model. Cross-lingual transfer of dependency parser have a long history. We encourage the reader to read a recent survey on this topic (Das and Sarkar, 2020). Our work does not use gold-standard parallel data or even supervised translation models to apply annotation projection.

7 Conclusion

We have described a fast and effective algorithm for learning translation systems using Wikipedia. We show that by wisely choosing what to use as seed data, we can have very good seed parallel data to mine more parallel text from Wikipedia. We have also shown that our translation models can be used in downstream cross-lingual natural language processing tasks. In the future, we plan to extend our approach beyond Wikipedia to other comparable datasets like the BBC World Service. A clear extension of this work is to try our approach on other cross-lingual tasks. Moreover, as many captions of the same images in Wikipedia are similar sentences and sometimes translations, multimodal machine translation (Specia et al., 2016; Caglayan et al., 2019; Hewitt et al., 2018; Yao and Wan, 2020) based on this data or the analysis of the data, such as whether more similar languages may share more similar captions (Khani et al., 2021) are other interesting avenues.
Acknowledgments

We would like to thank reviewers and the editor for their useful comments. We also would like to thank Alireza Zareian, Daniel (Joongwon) Kim, Qing Sun, and Afra Feyza Akyurek for their help and useful comments throughout this project. This work is supported in part by the DARPA LwLL Program (contracts FA8750-19-2-0201 and HR001118S0044), the DARPA KAIROS Program (contract FA8750-19-2-1004), the IARPA BET-TER Program (contract 2019-19051600004), and Department of the Air Force (contract FA8750-19-2-3334 - Semi-supervised Learning of Multimodal Representations). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA, the Air Force, and the U.S. Government.

References


Alex Jones and Derry Tanti Wijaya. 2021. Majority voting with bidirectional pre-translation for bitext retrieval.


### A Cross-Lingual Embedding

We use the off-the-shelf 300-dimensional FastText embeddings (*Grave et al., 2018*) as monolingual embedding vectors. We run FastAlign (*Dyer et al., 2016a*)...
We pick comparable candidates for sentence pairs and make use of the cross-lingual CCA tool \cite{faruqui2014improved} to extract 150-dimensional vectors. This tool can be run on a single CPU within a few hours.

\section*{B Monolingual and Translation Datasets}

We use an off-the-shelf Indic-transliteration library\footnote{https://pypi.org/project/indic-transliteration} to convert the Devanagari script to Hindi script to make the Hindi documents look like Gujarati by removing the graphical vertical bars from Hindi letters, thus increasing the chance of capturing more words in common. We boost the Romanian, Gujarati, and Kazakh monolingual data with newstext dataset from WMT. For parallel data in similar languages, we use the Mizan parallel data for Persian \cite{kashefi2018mizan} with one million sentences, the IITB data \cite{kunchukuttan2018pundit} and HindiEnCorp 0.5 \cite{bojar2014hindi} for Hindi with a total of 367K sentences, ParaCrawl for Russian \cite{espil2019paracrawl} with 12M sentences, and Europarl for Italian \cite{koehn2005europarl} with 2M sentences. We use the Arabic-English UN data \cite{ziemski2016common} for Arabic-English and Kazakh-English, and WMT 2016 shared task data \cite{bojar2016wmt} for Romanian-English. Following previous work \cite{sennrich2016improving}, diacritics are removed from the Romanian data.

\section*{C Translation Training Parameters}

We pick comparable candidates for sentence pairs whose lengths are within a range of half to twice of each other. As we see, the final size of mined datasets heavily depends on the number of paired English-target language Wikipedia documents. We train our translation models initialized by pretrained models. Each batch has roughly 4K tokens. Except for Arabic, for which the size of mined data significantly outnumbers the size of Persian-English parallel data, we use the related language data before using iterative back-translation in which we only use the source and target monolingual datasets. We use similar learning hyper-parameters to pretraining except for iterative back-translation in which we accumulate gradients for 100 steps, and use a dropout probability of 0.2 and 10000 warmup steps since we find smaller dropout and warmup make the model diverge. Our one-shot back-translation experiments use a beam size of 4, but we use a beam size of one for iterative back-translation since we have not seen much gains in using beam-based iterative back-translation except for purely unsupervised settings. All of our translations are performed with a beam size of 4 and $\text{max}_\text{len}_a = 1.3$ and $\text{max}_\text{len}_b = 5$. We alternate between supervised parallel data of a similar language paired with English and the mined data.

We train translation models for roughly 400K batches except for Gujarati that has smaller mined data for which we train for 200K iterations. We have seen a quick divergence in Kazakh iterative back-translation, thereby we stopped it early after running it for one epoch of all monolingual data. Most likely, the mined data for Kazakh-English has lower quality (see the supplementary material for more details), and that leads to very noisy translations in back-translation outputs. All of our evaluations are conducted using SacreBLEU \cite{post2018sentence} except for en-ro in which we use BLEU score \cite{papineni2002bleu} from Moses decoder scripts \cite{koehn2007moses} for the sake of comparison to previous work.

\section*{D Quality of Mined Data}

The quality of parallel data matters a lot for getting high-accuracy. For example, we manually observe
that the quality of mined data for all languages are very good except for Kazakh. Our hypothesis is that the Kazakh Wikipedia data is less aligned with the English content. We compare our mined data to that of the supervised mined data from WikiMatrix (Schwenk et al., 2019a) as well as gold-standard data. Figure 7 shows the difference between the three datasets of three language pairs (WikiMatrix does not contain Gujarati). As we see, our data has BLEU scores near to WikiMatrix in all languages, and in the case of Kazakh, the model trained on our data performs higher than WikiMatrix. In other words, in the case of having very noisy comparable data, as is the case for Kazakh-English, our model even outperforms a contextualized supervised model. It is also interesting to see that our model outperforms the supervised model for Kazakh that has only 7.7K gold-standard training data. These are all strong evidences of the strength of our approach in truly low-resource settings.

### E Pretraining Matters

It is a truth universally acknowledged, that a single model in possession of a small training data and high learning capacity, must be in want of a pre-trained model. To prove this, we run our translation experiments with and without pretraining. In this case, all models with the same training data and parameters are equal, but some models are more equal. Figure 8 shows the results on the mined data. Clearly, there is a significant gain by using pre-trained models. For Gujarati, which is our the lowest-resource language in our experiments, the distance is more notable: from BLEU score of 2.9 to 9.0. If we had access to a cluster of high-memory GPUs, we could potentially obtain even higher results throughout all of our experiments. Therefore, we believe that part of the blame for our results in English-Romanian is on pretraining. As we see in Figure 7, our supervised results without back-translation are also low for English-Romanian.

### F Comparing to CRISS

The recent work of Tran et al. (2020) shows impressive gains using high-quality pretrained models and iterative parallel data mining from a larger comparable data than that of Wikipedia. Their pretrained model is trained using 256 Nvidia V100 GPUs in approximately 2.5 weeks (Liu et al., 2020). Figure 9 shows that by considering all these facts, our model still outperforms their supervised model in English-to-Kazakh with a big margin (4.3 cs 10.8) and gets close to their performance in other directions. We should emphasize on the fact that Tran et al. (2020) explores a much bigger comparable data than ours. One clear addition to our work is exploring parallel data from other available comparable datasets. Due to limited computational resources, we skip this part but we do believe that using our current unsupervised models can help extract even more high-quality parallel data from comparable datasets, and this might lead to further gains for low-resource languages.