Automatic Evaluation and Analysis of Idioms in Neural Machine Translation

Christos Baziotis[*]
University of Edinburgh
c.baziotis@ed.ac.uk

Prashant Mathur
Amazon AI
pramathu@amazon.com

Eva Hasler
Amazon AI
ehasler@amazon.com

Abstract

A major open problem in neural machine translation (NMT) is the translation of idiomatic expressions, such as "*under the weather*". The meaning of these expressions is not composed by the meaning of their constituent words, and NMT models tend to translate them literally (i.e., word-by-word), which leads to confusing and nonsensical translations. Research on idioms in NMT is limited and obstructed by the absence of automatic methods for quantifying these errors. In this work, first, we propose a novel metric for *automatically* measuring the frequency of literal translation errors without human involvement. Equipped with this metric, we present *controlled* translation experiments with models trained in different conditions (with/without the test-set idioms) and across a wide range of (global and targeted) metrics and test sets. We explore the role of monolingual pretraining and find that it yields substantial *targeted* improvements, even without observing any translation examples of the test-set idioms. In our analysis, we probe the role of idiom context. We find that the randomly initialized models are more local or "myopic" as they are relatively unaffected by variations of the idiom context, unlike the pretrained ones.

1 Introduction

Neural machine translation (NMT; Sutskever et al. 2014; Bahdanau et al. 2015; Vaswani et al. 2017) struggles with the translation of rare multi-word expressions (MWE) (Koehn and Knowles, 2017). Non-compositional phrases, such as idioms (e.g., "piece of cake"), are one of the most challenging types of MWEs, because their meaning is figurative and cannot be derived from the meaning of their constituents (Nunberg et al., 1994; Liu, 2017). NMT models tend to translate these expressions literally (i.e., word-by-word), which leads to erroneous translations. In this paper, our focus is on the translation of idiomatic expressions, in contrast to most prior work, which is subsumed under MWEs in general (Constant et al., 2017; Cook et al., 2021).

The absence of *targeted* and *automatic* evaluation is a major obstacle to advances in idiom translation. Global metrics, such as BLEU (Papineni et al., 2002) consider the full translation, and thus, the effects of idiom translation are overshadowed. Previous efforts on targeted evaluation isolate the idiom translation using word alignments (Fadaee et al., 2018) or word edit distance (Zaninello and Birch, 2020). These approaches measure the accuracy of idiom translation but do not account for literal translation errors. Shao et al. (2018) proposed a method for estimating the frequency of such errors, but it requires the creation of language-specific hand-crafted lists (i.e., blocklists) with words that correspond to literal translation errors.

In this work[1], we present a study of idioms in NMT, with the goal of facilitating future research in this direction. First, we propose a novel metric for the *automatic* evaluation of literal translation errors (LitTER), that does not require any hand-crafted blocklists. We incorporate LitTER, which complements alignment-based metrics (Fadaee et al., 2018) into a unified targeted evaluation framework.

Next, we present translation experiments in a controlled setting, by using different training splits to test models under different conditions (e.g., zero-shot). To improve idiom translation we leverage monolingual data, which are more abundant than parallel and contain idioms in higher frequencies and more diverse contexts. We exploit monolingual data via pretraining (mBART; Liu et al. 2020), which is a generic and task-agnostic approach, unlike prior work that considers ad-hoc solutions (Fadaee et al., 2018; Zaninello and Birch, 2020). We find that monolingual pretraining yields strong targeted gains, even when models have not seen any translation examples of the test idioms.

[*] This work was done during an internship at Amazon.

[1] Code and data in github.com/amazon-research/idiom-mt

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3682–3700
May 2-6, 2023 ©2023 Association for Computational Linguistics

We also present an extensive analysis of how different models translate idioms. Specifically, we use a series of probing methods that encode idioms within different contexts (Garcia et al., 2021; Yu and Ettinger, 2020), and measure how this affects the translation outputs and the decoder distributions. We find that the randomly initialized models are more "myopic" compared to the pre-trained ones, as they are relatively unaffected when we vary the idiom context. Our contributions are:

1. We propose LitTER (§2.1), a novel metric for measuring the frequency of literal translation errors, and embed it into a framework (§2) for *automatic* and *targeted* evaluation of idiom translation, complementing prior work.
2. We present translation results (§3.3) in a *controlled* setting and across a wide range of metrics. We find that pre-training on monolingual data yields substantial *targeted* improvements.
3. We present an extensive *analysis* (§4) with a series of probes, showing how context affects idiom translation. We find that models are more uncertain when translating idioms and that pre-training makes models more contextual.

2 Automatic Targeted Evaluation

2.1 Literal Translation Error Rate (LitTER)

We propose literal translation error rate (LitTER), a novel metric of the frequency of literal translation errors made by a model. A literal translation error occurs if any of the words of a span in the source sentence has been *wrongly* translated literally in the target language. Our metric is inspired by the method of Shao et al. (2018) which identifies possible literal translation errors, by checking if a translation output contains any blocklisted words. While this method is effective at capturing these errors, it relies on hand-crafted blocklists. We overcome this limitation by automatically creating word blocklists for a given expression.

Our method, is based on two key ideas. First, we use bilingual word dictionaries[2], which are relatively easy to obtain, to translate the words of an annotated source span into the target language, and produce blocklists with candidate literal translation errors. Then, we use the reference translations to filter the blocklists by removing those words that occur in the reference. This avoids triggering the blocklist when the correct translation is literal.

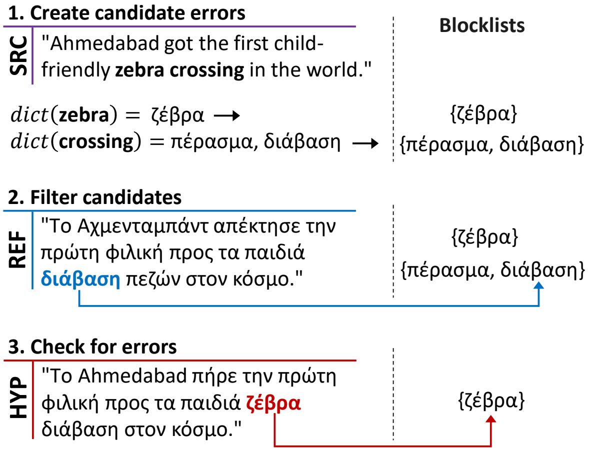

1. Create candidate errors
SRC
"Ahmedabad got the first child-friendly **zebra crossing** in the world."
dict(**zebra**) = ζέβρα →
dict(**crossing**) = πέρασμα, διάβαση →
Blocklists
{ζέβρα}
{πέρασμα, διάβαση}
2. Filter candidates
REF
"Το Αχμενταμπάντ απέκτησε την πρώτη φιλική προς τα παιδιά **διάβαση** πεζών στον κόσμο."
{ζέβρα}
{πέρασμα, διάβαση}
3. Check for errors
HYP
"Το Ahmedabad πήρε την πρώτη φιλική προς τα παιδιά **ζέβρα** διάβαση στον κόσμο."
{ζέβρα}

Figure 1: Overview of the algorithm for the Literal Translation Error Rate (LitTER). For each sentence, we first produce candidate literal translation errors (blocklist), using all the word translations of the source idiom words. Then, we filter the candidates in the blocklist by looking at the reference. Finally, we check if the hypothesis triggers the remaining words in the blocklist.

Algorithm

1. Select from the source text the list of words $\boldsymbol{s} = \langle s_1, s_2, ..., s_N \rangle$ that belong to the annotated expression (i.e., idiom).
2. For each word s_i, obtain all its word translation(s) in the target language using a bilingual word dictionary and add them to a blocklist $b_i = \langle t_1, t_2, \ldots, t_M \rangle$, creating a candidate *list* of blocklists $\boldsymbol{B_s} = \langle b_1, b_2, ..., b_N \rangle$.[3]
3. For each word in the reference (R), search if it occurs in any of the blocklists b_i. If so, remove the corresponding blocklist b_i from $\boldsymbol{B_s}$ to avoid false positives. For example in Figure 1, where words διάβαση and πέρασμα are synonyms, if we remove only διάβαση but leave πέρασμα as a blocklisted word and a model generates it in its translation, this will *wrongly* trigger a literal translation error.
4. Check if the hypothesis contains any blocklisted words. If it does, then we mark this hypothesis as having a literal translation error.

The final score is the percentage of translations that trigger the blocklist. As LitTER requires source-side annotations, we collect test data with idioms on the source side and annotate the spans where they occur (§3.1). Appendix C shows examples of LitTER evaluating real sentences in our data.

[2] In this work we use the MUSE (Lample et al., 2018).

[3] In practice, $t_1, t_2, \ldots, t_M$ in a blocklist are synonyms of each other as they are translations of the same source word.

2.2 Alignment-based Evaluation

To measure idiom translation accuracy, we use Alignment-based Phrase Translation Evaluation (APT-Eval), by extending Fadaee et al. (2018) with subword-level metrics. APT-Eval uses word alignments to find the words in the hypothesis and reference sentences, respectively, that align with the annotated idiom source span, and then compares the retrieved matches to each other. We consider two evaluation metrics. First, we use *unigram precision*, that measures the ratio of words in the reference spans that occur in the hypothesis spans, as in Fadaee et al. (2018). We also use ChrF (Popović, 2015), that measures character n-gram overlap.

LitTER vs. APT-Eval While APT-Eval is a targeted evaluation metric, it only measures translation accuracy. This means that given an inaccurate translation, it is impossible to measure whether it has a literal translation error. LitTER, however, quantifies this particular issue that affects NMT.

2.3 Handling Idiom Frequency Imbalances

Different idioms have significantly different frequencies (Appendix A.1). However, prior work has overlooked this fact (Zaninello and Birch, 2020; Fadaee et al., 2018; Shao et al., 2018; Rikters and Bojar, 2017). Thus, over-represented idioms can skew the reported results and favour models that have overfitted on them. To address this, we report all of our targeted evaluation results (i.e., LitTER, APT-Eval) by macro-averaging over idioms:

$$\boldsymbol{E}(\theta) = \frac{1}{|\boldsymbol{L}|}\sum_{j=1}^{|\boldsymbol{L}|}\frac{1}{|\boldsymbol{L_j}|}\sum_{i=1}^{|\boldsymbol{P}|}\boldsymbol{M}(\theta(s_i), t_i) \qquad (1)$$

where $\boldsymbol{L}$ denotes the set of distinct idioms in a test set and $\boldsymbol{P} = \{\langle s_i, t_i\rangle | L_j \in \langle s_i, t_i\rangle\}$ denotes the set of sentence pairs containing the idiom L_j. The model is denoted by θ and the translation of x by $\theta(x)$. We first compute the average score for the test pairs of each idiom with a given metric $\boldsymbol{M}$, and then average these values to produce $\boldsymbol{E}$.

3 Experiments

3.1 Data and Training Splits

We present experiments on en→fr and en→es data. For each language pair, we concatenate the data from Europarl v7[4] (Koehn, 2005), part of the WMT news translation task (Bojar et al., 2014), and from TED talk transcripts released as part of IWSLT 2017 shared task[5] (Cettolo et al., 2017).

Idiom Data We split the parallel data into regular and idiom data using a pattern-matching tool that we developed. Our tool takes as input a list of idioms and extracts sentences from a corpus containing these idioms. We also annotate the span in which each idiom occurs within a sentence, to enable the targeted evaluation metrics. This approach is similar to Fadaee et al. (2018), but we build our tool on top of Spacy's (Honnibal and Montani, 2017) rule-based matching engine. For each phrase in the input list, we automatically create pattern-matching rules that capture complex variations of a given phrase. See Appendix A for details.

In this work, we use a list of 225 English idioms, that we manually collected and make publicly available. We feed this list into our pattern-matching tool, and extract (and annotate) translation pairs that contain an idiom on the source side. The regular data are used *only* for training. The idiom data are further divided into the *idiom-train* and *idiom-test* sets. For each idiom (e.g., "under the weather") in our original idiom data, we put half of its sentence pairs to the idiom-train and the other half into the idiom-test sets, to obtain a balanced distribution. We discard sentences with idioms that occur only once. We conduct controlled experiments in the following testing conditions:

- **Zero**: training data includes only regular parallel data, and we measure how models perform on unseen idioms at test time.
- **Joint**: training data includes the regular and idiom-train data, and we measure how models perform on idioms observed (in a different context) in training data.
- **Upsampling**: same as the joint split, but we up-sample the idiom-train data N times. This setting measures whether it is necessary to upsample the targeted training data (idiom-train) to achieve better translation quality of idioms.

Evaluation For development, we use the IWSLT dev-set for each language pair. For general purpose translation evaluation, we report results in the WMT newstest14 and IWSLT'17 test sets for en→fr, and in the WMT newstest13 in particular and IWSLT'17 test sets for en→es. For the targeted idiom evaluation (i.e., LitTER and APT-Eval) we

[4] www.statmt.org/europarl/

[5] sites.google.com/site/iwsltevaluation2017/TED-tasks

Data	en→fr	en→es
Europarl	2,007,723	1,965,734
IWSLT	275,085	265,625
Combined (after preprocessing)	2,155,543	2,119,686
Regular	2,152,716	2,116,889
Idiom-train	1,327	1,312
Idiom-test	1,383	1,373
WMT-test	3,003	3,000
IWSLT-test	2,632	2,502

Table 1: Dataset statistics

use the extracted idiom-test data per language pair. To generate the word alignments for APT-Eval, we trained a fast-align (Dyer et al., 2013) model on each language-pair's training data. For decoding, we use beam search with beams of size 5, and evaluate all models using BLEU (Papineni et al., 2002) computed with SacreBLEU (Post, 2018).

Preprocessing We first filter out sentence pairs with more than 80 words or with length ratio over 1.5. Then, we tokenize the remaining sentences using sentencepiece[6](SPM; Kudo and Richardson 2018). For the randomly initialized models, we train SPM models with a joint vocabulary of 60K symbols on the concatenation of the source- and target-side of the regular training data. For the mBART fine-tuning experiments, we use the SPM model of mBART (250K symbols).

3.2 Models

Besides training models from scratch, we also investigate how pretraining on monolingual data affects idiom translation, which yields substantial improvements on generic translation quality (Lample and Conneau, 2019; Song et al., 2019; Liu et al., 2020). However, it is not obvious if monolingual data can help idiom translation, as they do not contain any examples with how to translate an idiom from one language into another.

We use mBART (Liu et al., 2020) via finetuning, which is pretrained on monolingual data from many languages. We hypothesize that one way multilingual pre-training can help is by bootstrapping over the source and target language contexts in which idioms occur. We also consider injecting different types of noise during fine-tuning, to corrupt the (encoder or decoder) input context and measure the effects on the targeted evaluation metrics. Specifically, we use source-side word masking and replacement (Baziotis et al., 2021), and target-side word-replacement noise (Voita et al., 2021). In our experiments, "random" denotes a randomly initialized model, while "mBART" stands for using mBART as initialization. For noisy finetuning we train the following variants: "mBART+mask" where we mask 10% of the source tokens, "mBART+replace (enc)" where we replace 10% of the source tokens with random ones, and "mBART+replace (dec)" where we replace 10% of the target tokens with random ones.

Model Configuration For fair comparison, the randomly initialized models use the same architecture as mBART. Specifically, the models are based on the Transformer architecture, with 12 encoder and decoder layers, 1024 embedding size and 16 self-attention heads. Our code is based on the official mBART implementation in Fairseq.

Optimization We optimized our models using Adam (Kingma and Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.999$, and ϵ=1e-6. For the random initialization experiments, the models were trained for 140K updates with batches of 24K tokens, using a learning rate of 1e-4 with a linear warm-up of 4K steps, followed by inverted squared decay. For the mBART initialization experiments, the models were trained for 140K updates with batches of 12K tokens, using a fixed learning rate of 3e-5 with a linear warm-up of 4K steps. In all experiments, we applied dropout of 0.3, attention-dropout of 0.1 and label smoothing of 0.1. For model selection, we evaluated each model every 5K updates on the dev set, and selected the one with the best BLEU.

3.3 Results

In this section, for brevity, we discuss a subset of our results, in particular our experiments in en→fr. Results for en→es are consistent with en→fr and are included in Appendix B. Table 2 summarizes all of our main results. Besides global evaluation using BLEU (§3.3.2) on diverse test sets, we also consider two targeted evaluation methods (§3.3.1) that focus on how the idioms are translated using our idiom-test set. For the upsampling split, we upsample the idiom-train data 20x. We also experimented with 100x upsampling, but models started to exhibit overfitting effects (see §B, §D).

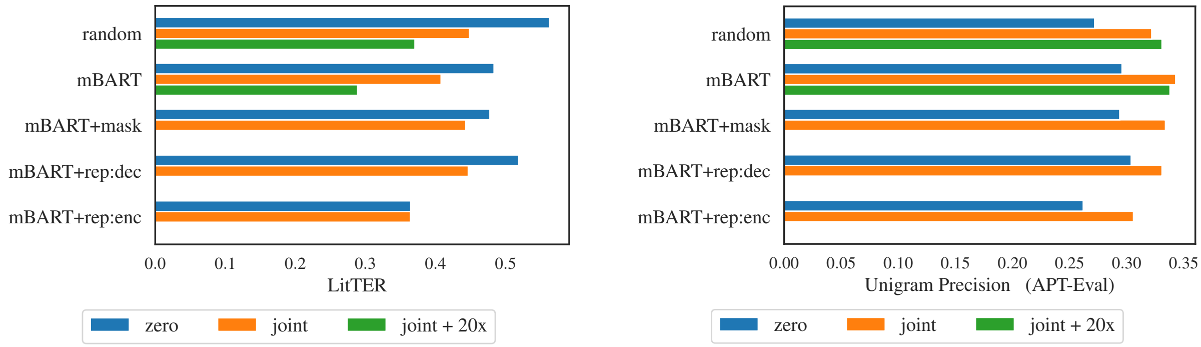
3.3.1 Targeted Evaluation

In targeted evaluation, we focus *only* on how models translate the source-side idioms. We present results on our proposed LitTER metric and on APT-

[6] We use the `unigram` model with `coverage=0.9999`

Split	Model (en→fr)	LitTER↓	APT-Eval		Global Evaluation (BLEU↑)		
			UniPrec↑	ChrF↑	IWSLT17	WMT14	Idiom-test
zero	random	0.563	0.268	0.298	44.1	34.8	34.4
	mBART	0.484	0.291	0.322	47.0	38.6	36.5
	mBART +mask	0.478	0.298	0.323	46.3	38.2	36.0
	mBART +replace (dec)	0.519	0.295	0.319	46.9	39.0	36.0
	mBART +replace (enc)	0.365	0.260	0.284	44.1	36.2	34.5
joint	random	0.448	0.317	0.337	44.2	34.8	35.3
	mBART	0.408	0.333	0.352	46.5	38.5	37.3
	mBART +mask	0.443	0.315	0.338	46.2	38.3	37.2
	mBART +replace (dec)	0.447	0.317	0.342	46.8	38.8	37.0
	mBART +replace (enc)	0.364	0.300	0.322	44.5	36.6	35.6
upsample 20x	random	0.371	0.323	0.353	44.4	34.7	35.3
	mBART	0.289	0.329	0.346	46.6	38.7	36.0

Table 2: All of our (en→fr) translation results (single run), including *generic* and *targeted* evaluation.



(a) Results on LitTER, which measures how often each model makes literal translation errors.

(b) Results on APT-Eval with unigram precision, which compares (the aligned) reference and hypothesis spans.

Figure 2: Results on *targeted* evaluation of idiom translation.

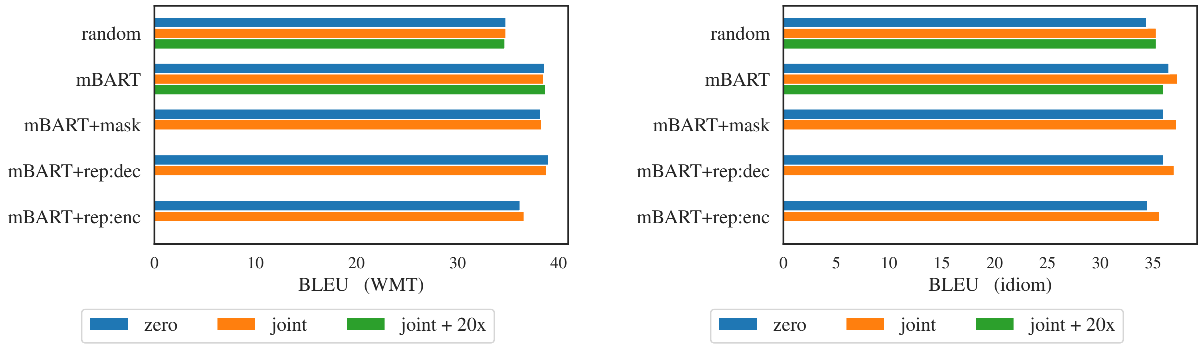
Eval, which provide different information. Recall that we macro-average these scores (§2.3) to account for imbalances in the idiom frequency.

Literal Translation Errors Figure 2a, shows the results on LitTER, that measures how often models make literal translation errors. As expected, all models produce fewer errors when trained on the joint split compared to the zero split. Pretraining gives a significant boost, even on the joint split. This shows that pretraining helps, even though the models have not seen any examples of how to translate the test-set idioms. Upsampling the idiom-train data helps all models regardless of initialization.

Each type of noise induces a different behaviour compared to the mBART model. Masking yields no effect on the zero split, but increases errors on the joint split. Baziotis et al. (2021), show that masking promotes copying, which we speculate it could lead to word-by-word translation and increase LitTER. Decoder-side word replacements yield a similar behaviour in terms of LitTER, which we hypothesize could push the decoder to rely more on the encoder, therefore encouraging word-by-word translation. By contrast, when we add word replacements in the encoder, it greatly reduces LitTER in both splits. This aligns with the findings of Baziotis et al. (2021), who show that source-side word replacements make the decoder less prone to copying (or "trusting") the encoder.

Idiom Translation Accuracy To estimate how accurately models translate idioms, we compare the reference and hypothesis matches that align to the source idiom words. Figure 2b, compares models using unigram precision, and the results are consistent across all APT-Eval metrics (Table 2).

Similar to the LitTER results, the joint split significantly improves idiom translation accuracy as well. Again, pretraining outperforms random initialization, even on the joint split. Upsampling, however, does not yield any consistent improvements. While source-side word replacements reduce literal translation errors, they also degrade idiom translation accuracy. Our hypothesis is that as the decoder becomes less reliant on the encoder,



(a) Regular BLEU results on the *generic* WMT14 test set

(b) Regular BLEU results on our *idiom-test* set.

Figure 3: Results on *global* evaluation (BLEU) on different test sets.

this induces more hallucinations.

3.3.2 Global Evaluation

Here, we discuss how models perform based on global translation evaluation using BLEU.

General Purpose Figure 3a, shows the results on the WMT14 test set, but we note that the results are generally consistent with the IWSLT17 test set (Table 2). The mBART intialized models, unsuprisingly, yield significantly better results than random initialization. As expected, there is no measurable difference between splits, not even when upsampling idioms, as both IWSLT17 and WMT14 are generic test sets. Noisy finetuning methods fail to improve results. Encoder-side word replacements even degrade overall performance, which aligns with the hypothesis that they induce hallucinations.

Idiom The results on our idiom-test set (Figure 3b) show that models perform consistently better when trained on the joint split. Global evaluation, however, considers the full sentences and the impact of idiom translation is overshadowed (Rikters and Bojar, 2017), which can be seen by the very small differences between splits. This prevents fine-grained comparisons between models and highlights the need for targeted evaluation.

4 Analysis

To further understand how models translate idioms, we present an extensive analysis with a series of probes, focused on the role of idiom context. Specifically, using the annotations in our idiom-test data, we encode the idiom words within different contexts (Figure 4) and measure how it affects the decoder distributions and the translation output. We consider (1) *full context*, in which we encode the idiom phrase within the whole input sentence, (2) *phrase-level context*, in which we encode the idiom phrase in isolation, (3) *word-level context*, in which we encode each idiom word independently. Our probes follow a similar approach to prior work that evaluate the idiomaticity of (pretrained) Transformer-based models (Garcia et al., 2021; Tayyar Madabushi et al., 2021) by measuring how idiom representations are affected by their context (Yu and Ettinger, 2020), but we extend these methods to analyze how (pretrained) NMT models translate idioms. For brevity, we discuss here our most important findings, focusing on random-vs-mBART initialization. However, we include the rest of our results in Appendix D for completeness.

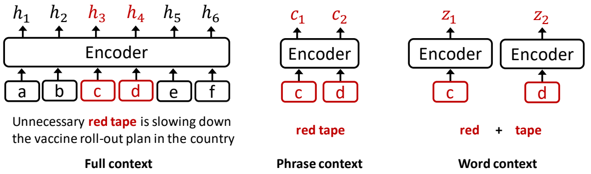


Figure 4: Illustration of how we obtain idiom representations by varying the available context.

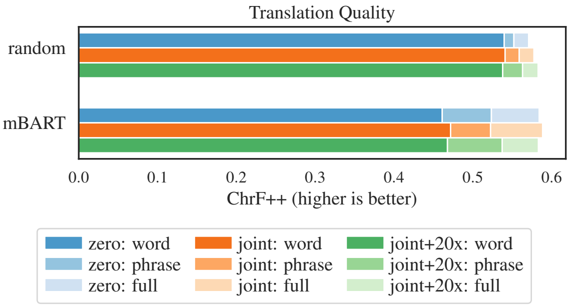


Figure 5: Variation in translation quality, measured in ChrF, as we vary the idiom representations. The length of each (lighter) bar encodes the difference from its (darker) bar to its left (i.e., overlapping bars effect).

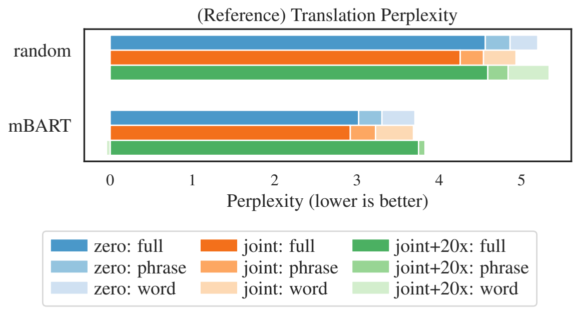


Figure 6: Perplexity of the references, as we vary the idiom representations. The length of each (lighter) bar encodes the difference from its (darker) bar to its left.

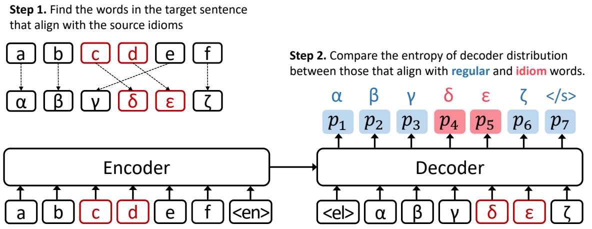

Step 1. Find the words in the target sentence that align with the source idioms
a
b
c
d
e
f
α
β
γ
δ
ε
ζ
Step 2. Compare the entropy of decoder distribution between those that align with regular and idiom words.
α
β
γ
δ
ε
ζ
</s>
p_1
p_2
p_3
p_4
p_5
p_6
p_7
Encoder
Decoder
a
b
c
d
e
f
<en>
<el>
α
β
γ
δ
ε
ζ

Figure 7: Illustration of how we compute the effects of different source idiom contexts on the model's uncertainty during translation (i.e., decoding).

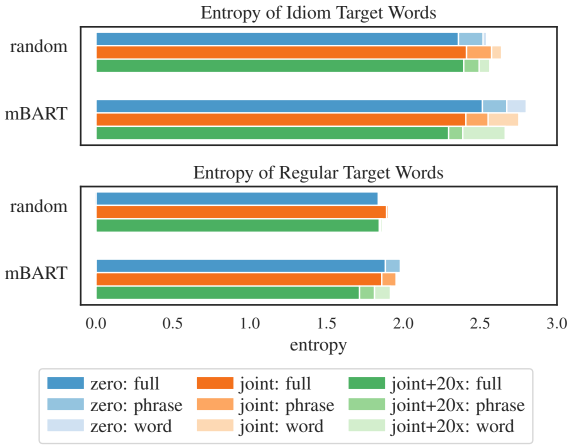


Figure 8: Model uncertainty for the translation of regular vs. idiom words. The length of each (lighter) bar encodes the difference from its (darker) bar to its left.

4.1 Variation in Translation Performance

In this probe, we decode (i.e., translate) different encoder representation and evaluate the samples against the reference (sentence) translations. Specifically, we first encode each (full) input sentence and then replace the encoder representations belonging to idiom words with those obtained with different (narrower) contexts. Figure 5 shows the results using ChrF, rather than BLEU, as a metric to capture even small subword-level changes.

Across all models, reducing the context (i.e., darker shades) results in worse translation scores. This is expected, as by swapping the original (full context) idiom representations with those obtained with word-context, we essentially remove information. When we use the full-context (i.e, lightest shades) pretraining yields the best results, but when we reduce the idiom context, the pretrained model suffers significantly, unlike the randomly initialized that is barely affected. This implies that the representations of the randomly initialized model are more local (or "myopic"), containing information mainly related to the idiom tokens. By contrast, the representations of the pretrained model are more global, and contain information related to the surrounding idiom context (Brunner et al., 2020). Upsampling does not have a strong effect.

4.2 Variation in Translation Likelihood

In this probe, we vary the encoder idiom representations, as before, and measure how this affects the likelihood of the reference translations. We score each reference translation by computing its perplexity under the model, given each encoder output sequence. Figure 6, shows the results, with *lighter* shades corresponding to *narrower* contexts.

Using more context improves (i.e., lowers) the perplexity of the references across models. Pretraining endows models with stronger LM capabilities which is probably why it yields generally lower perplexities. Training on the joint split yields consistent improvements which are more pronounced in the randomly initialized models. By contrast, upsampling the idiom-train data yields negative effects, which we attribute to overfitting, as it makes all other sentences less probable under the model.

4.3 Decoder Uncertainty

Next, we focus on how the token-level uncertainty of the decoder varies while it translates idiom vs. non-idiom words. First, we translate each sentence pair with teacher-forcing, where we feed to the decoder the reference translation (i.e., ground truth) as input, instead of its output from each step. Then, we measure the entropy of the decoder's distributions for each (reference) target token. Finally, using word alignments, we average separately the entropy values of target words[7] that are aligned to idiom and non-idiom source words. Figure 7 illustrates how the probe works. Figure 8 shows the

[7] After the word alignment step, we find which target SPM tokens correspond to which words.

results, in which *lighter* shades correspond to *narrower* contexts.

All models have significantly higher uncertainty when they translate idiom words (top section) compared to regular words (bottom section). This confirms our expectation that it is harder to translate idioms. When translating regular words, the randomly initialized models are unaffected by changes in the idiom representations, whereas reducing the idiom context increases the uncertainty of the pretrained models. This is another piece of evidence that pretraining yields less local models (§4.1). When translating idiom words, including or upsampling the idiom-train data benefits the pretrained model, but not the randomly initialized one.

5 Discussion

Global Metrics LiTER does not aim to replace global evaluation metrics like BLEU, but to complement them. Global metrics estimate the general translation quality of model outputs, which is undoubtedly important. However, they consider the full sentence, and as result the effects of idiom (mis)translation are overshadowed (see §3.3.2). LiTER aims to fill this gap by providing additional insights to practitioners with targeted evaluation.

LitTER LiTER should not be used in isolation, but combined with other (global/targeted) metrics. The reason is that lower LiTER can be achieved by more accurate idiom translations or by hallucinations. We aim to enable practitioners to make informed decisions without running human evaluations in the model development phase. The goal is to produce models that improve on LitTER without sacrificing general translation quality (e.g., BLEU). Our experiments reveal this contrast, where unsupervised pretraining achieves this goal, whereas (some of) the noisy variants fail. We expect LiTER to be more relevant when developing NMT models for creative content (e.g., subtitles, social media text) that usually contains figurative language.

Alignment Metrics Word alignment-based methods aim to capture idiom translation accuracy that complements LitTER. However, they are sensitive to the literal meaning of words to produce the alignments. Thus, while alignment-based methods could be reliable in certain types of evaluation, such as gender translation (Stanovsky et al., 2019), we believe that with the current techniques they should be used with caution for idiom translation evaluation. Although we did not systematically study this issue, we discovered by manual inspection that it was not uncommon to produce noisy or empty alignments. We chose statistical over embedding-based methods as they yielded less empty alignments (§B.2).

6 Related Work

Idioms in NMT There is limited research on idioms in NMT. Zaninello and Birch (2020), explore augmenting the training data with MWE translations from dictionaries, backtranslating (Sennrich et al., 2016) target-side sentences with MWEs, or wrapping the (source) MWEs with special tokens. Fadaee et al. (2018) prepend a special token in source sentences that contain an idiom. Gamallo and Garcia (2019) do unsupervised translation of MWEs by composing cross-lingual word embeddings, but this is fundamentally incapable of translating idioms which are non-compositional. Instead of using ad-hoc solutions that change the model or the data pipeline, we use monolingual pretraining, which is a more generic and less invasive approach.

Targeted Evaluation Using word alignments is a straightforward approach for the targeted evaluation of words or phrase translation (Stanovsky et al., 2019). Fadaee et al. (2018), use word alignments to compare the reference and hypothesis matches that translate a given source idiom. Zaninello and Birch (2020) first align words in the hypothesis and the reference using edit-distance and then score the aligned words using character-level matching. Alignment-based methods capture idiom translation accuracy, but they do not account for literal translation errors which are a major issue in idiom translation (Fadaee et al., 2018). The method of Shao et al. (2018) is capable of estimating the frequency of such errors, but it requires the creation of language-specific hand-crafted lists. In this work, we lift this limitation and enable the automatic evaluation of literal translation errors.

Analysis Rikters and Bojar (2017) investigate the attention mechanism in NMT during the translation of MWEs. Garcia et al. (2021) evaluate the idiomaticity of Transformer-based models using probes that measure how idiom representations are affected by their context (Yu and Ettinger, 2020). Similarly, Tayyar Madabushi et al. (2021) investigate the idiomaticity of pretrained encoders, such as BERT (Devlin et al., 2019), in a *monolingual* setting. By contrast, we present an analysis of how (pretrained) NMT models *translate* idioms.and

Concurrent analysis works explore idioms in NMT, by also using the blocklist method (Shao et al., 2018) as part of their analysis. Dankers et al. (2022a) study the compositionality of (Transformer) NMT models and (among others) find that models trained on more (parallel) data are more compositional. This relates to the results of our analysis (§4), which shows that pretrained models are less local than those trained only on the downstream parallel data. Dankers et al. 2022b analyze the hidden states and attention patterns of Transformer NMT models when processing idioms.

7 Conclusions

We present a comprehensive study of idiomatic expressions in NMT, aiming to facilitate future research on the topic. We propose LitTER (§2.1), a novel metric that enables the *automatic* evaluation of literal translation errors. LitTER is used for *targeted* evaluation and aims to complement and not replace global metrics, such as BLUE or ChrF, which consider the full sentence and can only measure the *overall* translation quality. We evaluate models in controlled conditions, with and without the test set idioms (i.e., zero-shot). We explore pretraining on monolingual data for improving idiom translation, as parallel idiom data is difficult to come by. Interestingly, we find that pretraining achieves strong targeted improvements, even in the zero-shot setting (§3.3.1). We also present a systematic analysis (§4) that investigates the role of context in idiom translation. We find evidence that pretraining yields more contextual models, which helps to explain why it contributes to better idiom translations. We also quantitatively confirm that idioms are more difficult to translate than regular words and strongly depend on the source context.

Limitations

LitTER is a novel metric that sidesteps the need for human involvement to estimate the frequency of literal translation errors, which is a major problem in idiom translation by NMT models. However, in its current iteration it has certain limitations:

1. An edge case with LitTER is that if the blocklist words appear as a result of translating other words - not part of the non-literal phrase - we will still count it as an error. We leave this as future work.

2. Our experiments were conducted on languages with relatively simple morphology. Preliminary experiments with German revealed that LitTER struggles with compound words. We did address these issues with custom rules, but we aim to study these cases in future versions of LitTER systematically.

3. The metric can be ambiguous when used in isolation. We recommend pairing it with standard evaluation metrics when comparing translation models.

Acknowledgments

We thank Marcello Federico, Georgiana Dinu, Anna Currey and Hoang Cuong for the helpful discussions and feedback during this project.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA. 1

Christos Baziotis, Ivan Titov, Alexandra Birch, and Barry Haddow. 2021. Exploring unsupervised pretraining objectives for machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2956–2971, Online. Association for Computational Linguistics. 4, 5

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics. 3

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers. In *International Conference on Learning Representations*. 7

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the iwslt 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–14. 3

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Survey: Multiword expression processing: A Survey. *Computational Linguistics*, 43(4):837–892. 1

Paul Cook, Jelena Mitrović, Carla Parra Escartín, Ashwini Vaidya, Petya Osenova, Shiva Taslimipoor, and Carlos Ramisch, editors. 2021. *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*. Association for Computational Linguistics, Online. 1

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022a. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175, Dublin, Ireland. Association for Computational Linguistics. 9

Verna Dankers, Christopher Lucas, and Ivan Titov. 2022b. Can transformer be too compositional? analysing idiom processing in neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3608–3626, Dublin, Ireland. Association for Computational Linguistics. 9

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 8

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics. 14

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics. 4, 14

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2018. Examining the tip of the iceberg: A data set for idiom translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). 1, 3, 8, 12

Pablo Gamallo and Marcos Garcia. 2019. Unsupervised compositional translation of multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 40–48, Florence, Italy. Association for Computational Linguistics. 8

Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564, Online. Association for Computational Linguistics. 2, 6, 8

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear. 3, 12

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA. 4

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand. 3

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics. 1

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics. 4

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*. 4

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*. 2

Dilin Liu. 2017. *Idioms: Description, comprehension, acquisition, and pedagogy*. Routledge. 1

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742. 1, 4

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. In Stephen Everson, editor, *Language*, pages 491–538. Cambridge University Press. 1

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics. 1, 4

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics. 3, 13

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics. 4

Matīss Rikters and Ondřej Bojar. 2017. Paying attention to multi-word expressions in Neural Machine Translation. In *Proceedings of MT Summit XVI. Main Conference: Research Track* , volume 1, pages 86–95. 3, 6, 8

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics. 8

Yutong Shao, Rico Sennrich, Bonnie Webber, and Federico Fancellu. 2018. Evaluating machine translation performance on Chinese idioms with a blacklist method. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). 1, 2, 3, 8, 9

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pretraining for language generation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR. 4

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics. 8

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112. 1

Harish Tayyar Madabushi, Edward Gow-Smith, Carolina Scarton, and Aline Villavicencio. 2021. AStitchInLanguageModels: Dataset and methods for the exploration of idiomaticity in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3464–3477, Punta Cana, Dominican Republic. Association for Computational Linguistics. 6, 8

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 5998–6008, Long Beach, CA, USA. 1

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1126–1140, Online. Association for Computational Linguistics. 4

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Online. Association for Computational Linguistics. 2, 6, 8

Andrea Zaninello and Alexandra Birch. 2020. Multiword expression aware neural machine translation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3816–3825, Marseille, France. European Language Resources Association. 1, 3, 8

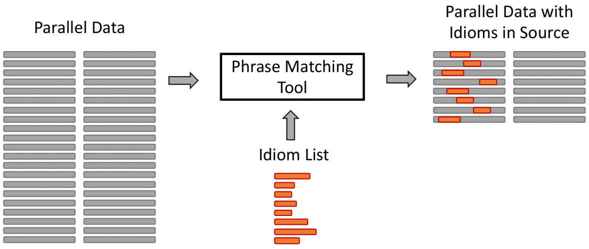
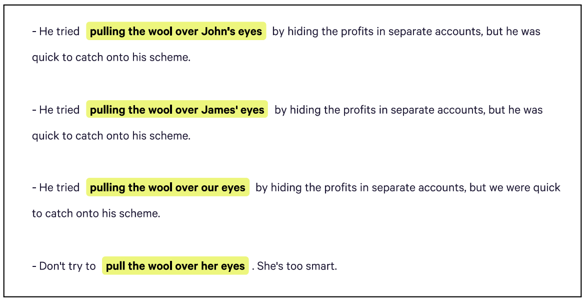


Figure 9: Overview of the process for collecting the idiom parallel data.



- He tried **pulling the wool over John's eyes** by hiding the profits in separate accounts, but he was quick to catch onto his scheme.

- He tried **pulling the wool over James' eyes** by hiding the profits in separate accounts, but he was quick to catch onto his scheme.

- He tried **pulling the wool over our eyes** by hiding the profits in separate accounts, but we were quick to catch onto his scheme.

- Don't try to **pull the wool over her eyes** . She's too smart.

Figure 10: Overview of the process for collecting the idiom parallel data.

A Idiom Data Collection

We collected parallel data with idioms in the source side and annotated the spans in which the idioms occur within each sentence. This enables us to conduct controlled experiments and to support our targeted evaluation metrics and our analysis. To collect the data, we created a phrase-matching tool that searches for idioms in parallel data and extracts and annotates the retrieved pairs (Figure 9).

Phrase-Matching Tool Our tool, which we make publicly available, uses rule-based matching to search for sentences that contain phrases specified in user-defined phrase list. While in this work we use it to create parallel data with idioms, it could be used to create datasets with different types of phrases. We build our tool on top of Spacy's (Honnibal and Montani, 2017) rule-based matching engine[8] that is more flexible and easy to work with than using regular expressions or custom rules (Fadaee et al., 2018). It allows us to do pattern matching over linguistic units, such as parts-of-speech or even dependency relations, thus capturing complex variations of a given phrase. Our tool first reads the input phrase list, and for each phrase, it automatically creates a pattern based on some simple rules and assumptions. We created separate idiom train and test data for each language-pair and

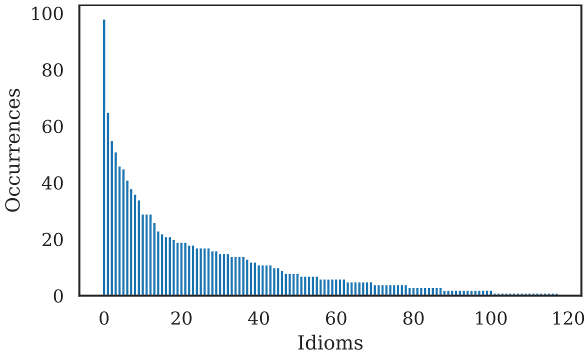


Figure 11: Occurrences of idioms in the en→fr idiom-test sets.

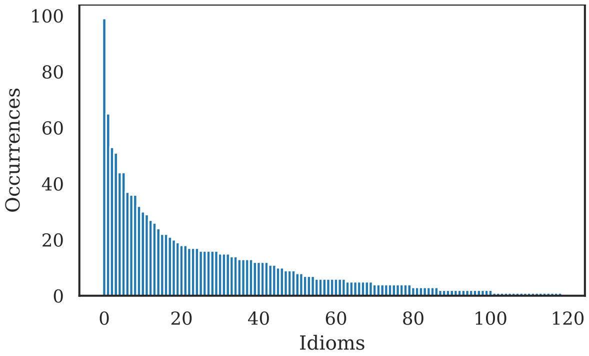


Figure 12: Occurrences of idioms in the en→es idiom-test sets.

we describe the process in Section 3.1.

Figure 10, shows an actual example[9] of the different variations of the idiom "pull the wool over someone's eyes" that our tool captures. Notice that it matches:

- Different variations of the verb "pull".
- Different words in the place of the word "someone".
- Importantly, it optionally matches the particle "'s". This shows that we can apply logic base on the part-of-speech (POS) or other linguistic properties of words, and in this case optionally skip them.

A.1 Idiom-test Statistics

Figure 11 and Figure 12 show the occurrences of idioms in the en→fr and en→es idiom-test sets, respectively. We see that in both test sets the idiom statistics follow very similar distributions. This verifies that different idioms have very different

[8] spacy.io/usage/rule-based-matching

[9] You can view this example in this online demo. It shows the rule we generate for the idiom phrase and the variants it captures.

Split	Model (en→fr)	LitTER↓	APT-Eval (fast-align)		APT-Eval (awesome-align)		Global Evaluation BLEU↑			Global Evaluation ChrF↑		
			UniPrec↑	ChrF↑	UniPrec↑	ChrF↑	IWSLT17	WMT14	idiom	IWSLT17	WMT14	idiom
zero	random	0.563	0.268	0.298	0.272	0.319	44.1	34.8	34.4	0.67	0.61	0.60
	mBART	0.484	0.291	0.322	0.296	0.347	47.0	38.6	36.5	0.69	0.64	0.61
	mBART +mask	0.478	0.298	0.323	0.294	0.339	46.3	38.2	36.0	0.68	0.64	0.61
	mBART +replace (dec)	0.519	0.295	0.319	0.304	0.346	46.9	39.0	36.0	0.69	0.64	0.61
	mBART +replace (enc)	0.365	0.260	0.284	0.262	0.306	44.1	36.2	34.5	0.66	0.62	0.60
joint	random	0.448	0.317	0.337	0.322	0.365	44.2	34.8	35.3	0.67	0.61	0.61
	mBART	0.408	0.333	0.352	0.343	0.384	46.5	38.5	37.3	0.68	0.64	0.62
	mBART +mask	0.443	0.315	0.338	0.334	0.379	46.2	38.3	37.2	0.68	0.64	0.62
	mBART +replace (dec)	0.447	0.317	0.342	0.331	0.379	46.8	38.8	37.0	0.69	0.64	0.62
	mBART +replace (enc)	0.364	0.300	0.322	0.306	0.350	44.5	36.6	35.6	0.66	0.62	0.61
upsample 20x	random (20x)	0.371	0.323	0.353	0.331	0.382	44.4	34.7	35.3	0.67	0.61	0.61
	mBART (20x)	0.289	0.329	0.346	0.338	0.376	46.6	38.7	36.0	0.68	0.64	0.61
upsample 100x	random (100x)	0.378	0.314	0.343	0.325	0.363	43.8	34.7	33.9	0.66	0.61	0.60
	mBART (100x)	0.289	0.337	0.358	0.347	0.389	46.8	38.2	35.8	0.69	0.64	0.61

Table 3: Translation results in en→fr. The results involve a single run, but include mutliple test sets and are consistent across the board.

Split	Model (en→es)	LitTER↓	APT-Eval (fast-align)		APT-Eval (awesome-align)		Global Evaluation BLEU↑			Global Evaluation ChrF↑		
			UniPrec↑	ChrF↑	UniPrec↑	ChrF↑	IWSLT17	WMT13	idiom	IWSLT17	WMT13	idiom
zero	random	0.541	0.350	0.364	0.351	0.370	36.0	31.5	38.9	0.62	0.57	0.63
	mBART	0.476	0.383	0.385	0.368	0.390	38.5	34.0	40.8	0.64	0.59	0.64
	mBART +mask	0.481	0.369	0.378	0.354	0.380	38.8	34.0	40.5	0.64	0.59	0.64
	mBART +replace (dec)	0.508	0.388	0.389	0.384	0.401	38.6	34.4	40.4	0.64	0.59	0.64
	mBART +replace (enc)	0.389	0.334	0.345	0.323	0.351	37.0	32.1	39.0	0.62	0.57	0.62
joint	random	0.468	0.385	0.395	0.382	0.397	35.9	31.8	39.7	0.62	0.57	0.64
	mBART	0.412	0.399	0.406	0.393	0.419	38.8	33.9	41.9	0.64	0.59	0.65
	mBART +mask	0.418	0.389	0.402	0.388	0.410	38.7	33.9	41.9	0.64	0.59	0.65
	mBART +replace (dec)	0.443	0.402	0.408	0.402	0.414	38.5	33.9	41.7	0.64	0.59	0.65
	mBART +replace (enc)	0.352	0.352	0.372	0.355	0.378	36.7	32.2	39.7	0.62	0.57	0.63
upsample 20x	random (20x)	0.400	0.410	0.424	0.415	0.440	35.9	31.7	40.0	0.62	0.57	0.64
	mBART (20x)	0.301	0.422	0.435	0.419	0.444	38.8	34.0	40.6	0.64	0.59	0.64
upsample 100x	random (100x)	0.391	0.406	0.420	0.407	0.435	36.2	31.8	38.9	0.62	0.57	0.63
	mBART (100x)	0.299	0.416	0.427	0.406	0.441	38.7	33.9	40.5	0.64	0.59	0.64

Table 4: Translation results in en→es. The results involve a single run, but are consistent across the board.

frequencies, and highlights the need for macro-averaging the scores of targeted evaluation metrics. Failure to do so would promote models that have perform best on the most frequent idioms over those that have a more balanced performance.

Also, note that the frequencies of idioms in the idiom-train and idiom-test sets are identical, as described in Section 3.1. While we have not computed the idiom frequencies in the monolingual data used to pretrain mBART, we expect that they would follow a similar distribution as the one found in the monolingual data we used in our work.

B Translation Results

In this section, we present all of our translation results in detail. Table 3, shows our results in en→fr, while Table 4 shows our results in en→es. Besides BLEU, we also include results with ChrF (Popović, 2015), for global translation evaluation. overall the results are *consistent* across language pairs in all evaluation methods. The fact that the absolute scores reached by the model on each each test set are different is natural, as the test sets themselves are different to each other between languages. However, the relative performance between model across language pairs is the same, with only minor differences.

B.1 Targeted Evaluation

In Figure 13, we visualize how model perform on our targeted evaluation methods for both en→fr and en→es. Recall that, LitTER, measures how often each model makes literal translation errors. APT-Eval, measures idiom translation accuracy, by comparing the reference and (aligned) hypothesis spans that translate the source idiom words. In these plots, we use present how models performed on unigram precision metric for APT-Eval, similar to the main paper. We observe that both in terms of literal translation errors (Figure 13a, 13c), as well as idiom translation accuracy (Figure 13b, 13d), the results are remarkably consistent across languages. Upsampling the idiom-train data helps all models regardless of initialization, but upsampling more than 20x does not yield consistent improvements.

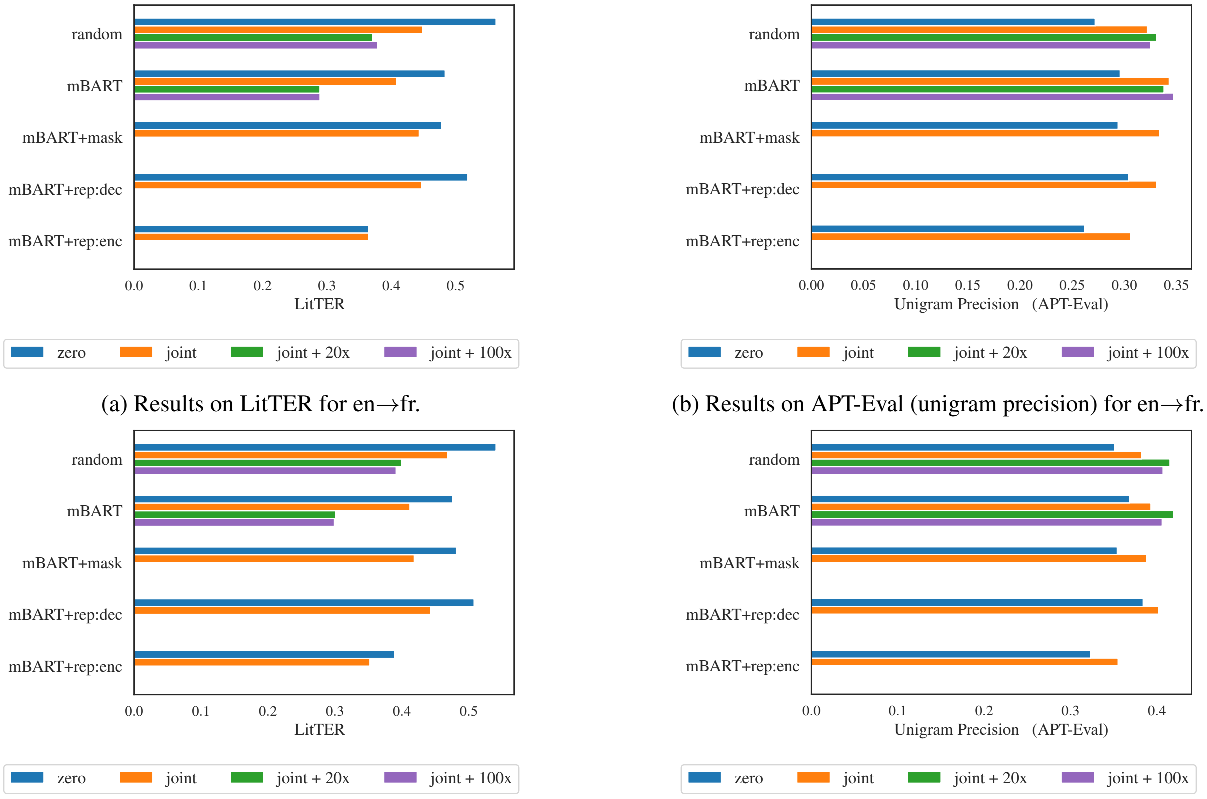
B.2 Word Alignment Models

APT-Eval, requires word alignments to map the source idiom words to the words of the reference and hypothesis spans, respectively. In our main paper, we presented results using fast-align (Dyer et al., 2013) with word alignment models trained on the training data of each language-pair. We also experimented with awesome-align (Dou and Neubig, 2021), that doesn't require any training, and uses the token similarities of the pretrained mBERT models to obtain the alignments. After analysis, we found that fast-align produced empty matches for the idiom words in 2.2% of the reference sentences in our idiom-test set. Awesome-align, however, yielded more empty matches, which after tweaking its threshold parameter[10], we managed to reduce it to 3%. While we omitted the awesome-align results from the main paper, we include them in Tables 3, 4 for completeness. We observe that while the absolute APT-Eval scores are different between the two alignment methods, the relative performance across models is consistent.

B.3 Regular Translation Evaluation

Here, we visualize our results on *regular* MT evaluation for both en→fr and en→es. In Figure14, we compare models in both language pairs and for both generic test sets as well as on our idiom-test sets. Overall, we observe that the results are very consistent between language pairs, similar to the targeted evaluation results (§B.1), which improves our confidence in them. As we already noted in our paper, we find that including or upsampling idiom training data has no measurable effect on generic test sets, unlike on our idiom-test set.

[10] We used the following hyper-parameters: `extraction=softmax, softmax_threshold=0.001`



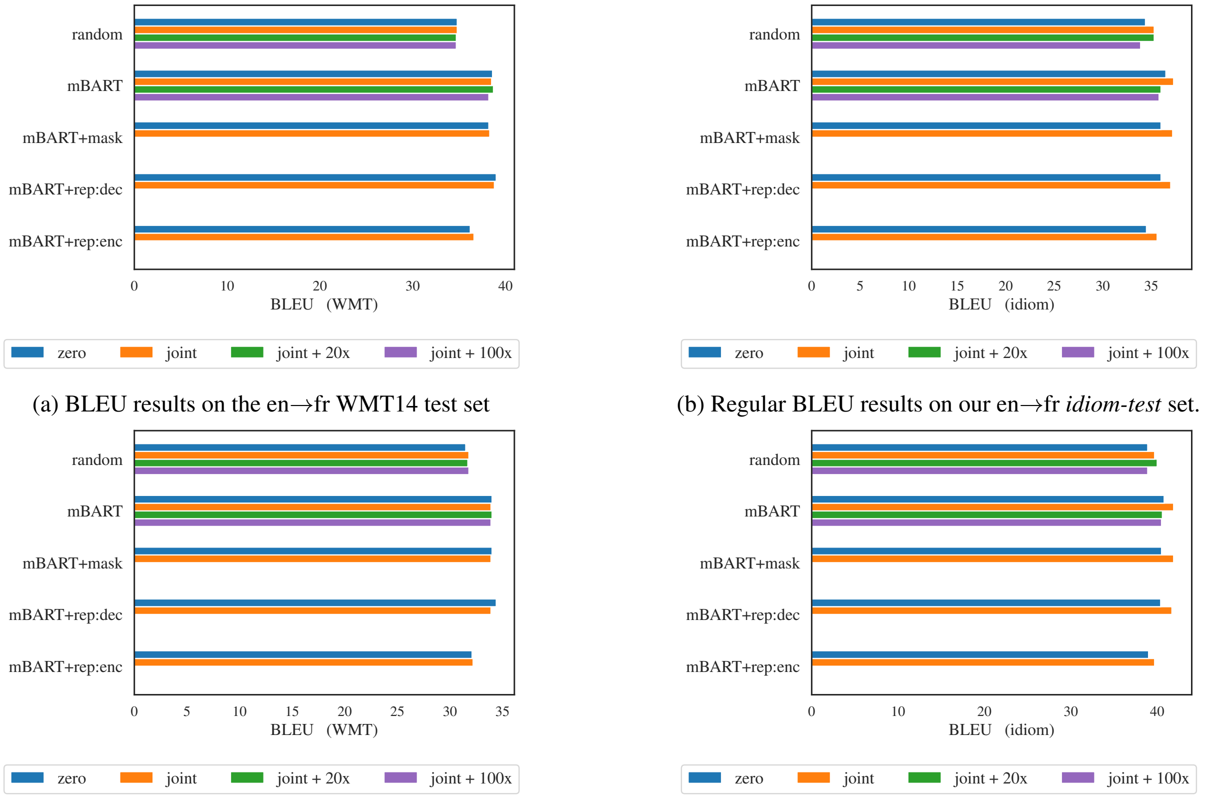
(a) Results on LitTER for en→fr.

(b) Results on APT-Eval (unigram precision) for en→fr.

(c) Results on LitTER for en→es.

(d) Results on APT-Eval (unigram precision) for en→es.

Figure 13: Results on *targeted* evaluation of idiom translation.



(a) BLEU results on the en→fr WMT14 test set

(b) Regular BLEU results on our en→fr *idiom-test* set.

(c) BLEU results on the en→es WMT13 test set

(d) Regular BLEU results on our en→es *idiom-test* set.

Figure 14: Results on *regular* MT evaluation with BLEU on generic as well as on our idiom-test sets.

C LitTER

C.1 Indicative Examples

Here, we present some indicative examples of how LitTER evaluates translation outputs in our idiom test data. Figure 16, contains examples where LitTER is working as intended, whereas Figure 15, contains a two failures of LitTER.

SRC: As the example of Cyprus shows, Ankara does not pull its punches.
REF: Comme le montre l'exemple de Chypre, Ankara n'y va pas avec le dos de la cuiller.
HYP: Comme le montre l'exemple de Chypre, Ankara ne tire pas les ficelles.

Blocklists
pull → {tirez, tirer}
its → {ses, son, sa}
punches → {coups}

No error detected.

SRC: [..] it was already being put on ice on the grounds that 'We'll never get it though the G20'.
REF: [..] elle était mise au rencart au motif que "nous n'arriverons jamais à convaincre le G20".
HYO: [..] on l'a déjà gelé au motif que "nous n'y arriverons jamais par le biais du G20".

Blocklists
put → {mis, mettre}
on → {sur}
ice → {glace, ice, verglas}

No error detected.

Figure 15: LitTER failures on our en→fr idiom test set. In the first example, the blocklist is not triggered because the inflected form *tire* is missing from the blocklist. In the second example, the verb form *gelé* (freeze) is not contained in the blocklist but is a literal translation of *ice* in a wider sense.

SRC: To postpone this vote one more time would be to bark up the wrong tree.
REF: Postposer ce vote une fois de plus eut été se tromper de cible.
HYP: Reporter ce vote une fois de plus, c'est se tromper d'arbre.

Blocklists
bark→ {aboyer, ecorces, ecorce}
up→ {debout}
the→ {le, la, les}
wrong→ {faux, tort, errone, mal}
tree→ {arbre, arbres, sapin, arborescence}

ERROR: Blocklist triggered by {arbre}

SRC: For companies, using technology to gather important data, its like bread and butter.
REF: Pour les sociétés, utiliser la technologie pour recueillir des données, c'est la routine.
HYP: Pour les entreprises, utiliser la technologie pour collecter des données importantes, c'est comme du pain et du beurre.

Blocklists
bread→ {pain}
and→ {et}
butter→ {et, pain, beurre}

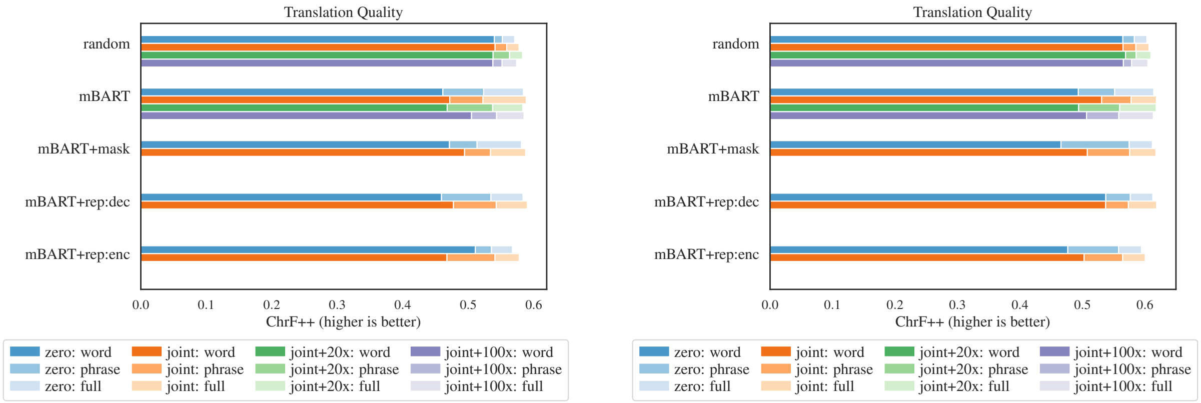
ERROR: Blocklist triggered by {et, pain, beurre}

SRC: And here is some eye candy for you, from a range of DIY scientists and artists from all over the globe.
REF: Et voici quelques bonbons pour vos yeux, de la part d'un éventail de scientifiques et des artistes bricoleurs de tous les coins de la planète.
HYP: Et voici quelques bonbons pour les yeux, d'une gamme de scientifiques et d'artistes du bricolage du monde entier.

Blocklists
eye → {oculaire, oeil, yeux, œil}
candy → {bonbon, bonbons, sucrerie}

No error detected.

Figure 16: Examples of LiTER evaluation on sentences in our en→fr idiom test set. In the first two examples, the model makes a literal translation error and the error is captured by LitTER. In the third example, the literal translation is correct and the blocklist is not triggered, thanks to the 3rd step in our algorithm (§ 2.1).



(a) Results for the en→fr models.

(b) Results for the en→es models.

Figure 17: Variation in translation quality, measured in ChrF, as we vary the idiom representations. The length of each (lighter) bar encodes the difference from its (darker) bar to its left (i.e., overlapping bars effect). The darkest shades correspond to using the full (original) context for encoding idiom words, and each lighter shades correspond to narrower contexts. Overall, the results are consistent, with the exception of the "mBART+replace(enc)" model.

D Analysis

In this section we present all of our analysis results. Specifically, we include results on en→es, with the (mBART) pretrained models finetuned with different noising methods, and additional probes. For the noisy versions of mBART finetuning, we present results on the zero and joint split. Recall that, in most of our probes, we evaluate the role of (idiom) context idiom translation. To do this, we encode the idiom words within different contexts and compare how this affects various aspects of each model. Figure 4, illustrates the process by which we obtain the encoding for each context. We consider the following context: (1) *full context*, in which we encode the idiom phrase within the whole input sentence, (2) *phrase-level context*, in which we encode together only the words idiom phrase, (3) *word-level context*, in which we encode each idiom word independently.

Figure 18, shows a visual example of how this probe works.

D.1 Variation in Translation Performance

With this probe, we test how the variation of the idiom encoder representations is reflected in the translation output. Figure 18, shows a visual example of how this probe works. First we encode each input sentence and then replace the encoder output representations belonging only to idiom words with those obtained with different (narrower) contexts. Finally, we decode each encoder output sequence and compare the generated translation to the reference translation.

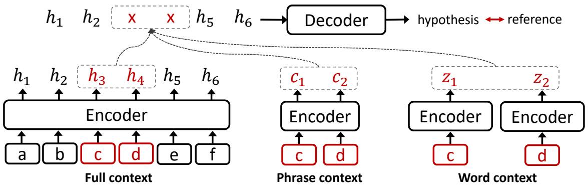
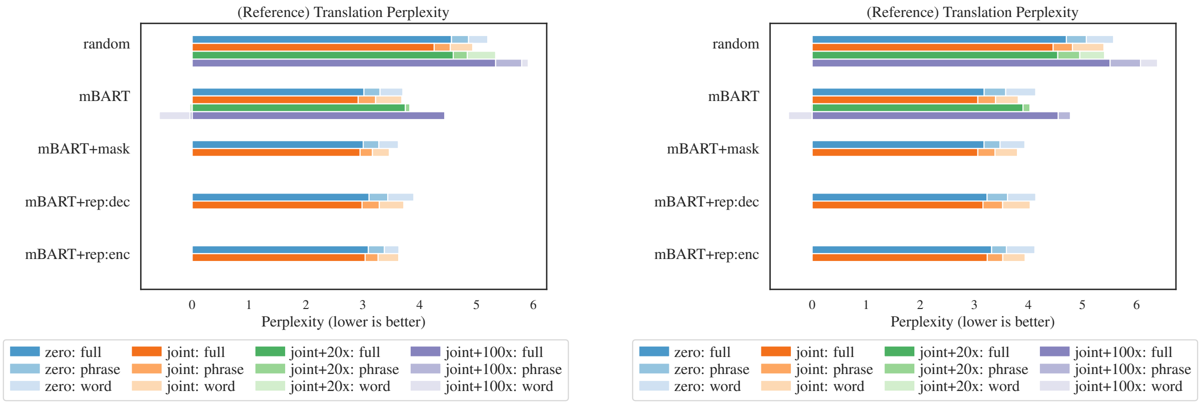


Figure 18: Illustration of how we probe the impact of different idiom contexts on translation quality. First, we obtain each sequence of idiom token representations, by encoding them within different contexts. Then we feed each sentence to the encoder, and before passing its outputs to the decoder, we replace the idiom token representations with those whose context we want to probe. Finally, we sample a translation from the decoder and compare it against the reference.

We observe that when using the full context, the results are generally consistent across language pairs and models. As we discussed in the main paper, the mBART-initialized model suffers greatly when the idiom representations are encoded with narrower context, in contrast to the randomly initialized model. We believe this is an indication that the mBART-initialized model is less local, meaning that each token representation contains to a large degree information about the rest of the tokens. We also observe that this behaviour is exhibited by all pretrained models.

However, while results are generally consistent in both language pairs, we do see some *small* discrepancies as we probe the effects of narrower contexts, in particular for the "mBART+replace(enc)" model. We do not have a satisfying explanation for this performance difference, which occurs only



(a) Results for the en→fr models.

(b) Results for the en→es models.

Figure 19: Variation in perplexity of reference translations, as we vary the idiom representations. The length of each (lighter) bar encodes the difference from its (darker) bar to its left. Negative bar lengths indicate a decrease relative to the (darker) bar before it. The darkest shades correspond to using the full (original) context for encoding idiom words, and each lighter shades correspond to narrower contexts.

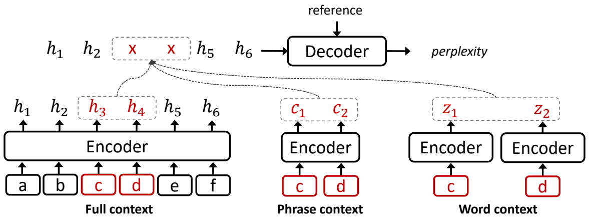


Figure 20: Illustration of how we measure the effects of different idiom contexts on translation likelihood (perplexity). First, we obtain each sequence of idiom token representations, by encoding them within different contexts. Then we feed each sentence to the encoder, and before passing its outputs to the decoder, we replace the idiom token representations with those whose context we want to probe. We use the sequence of encoder outputs, to score the reference translation.

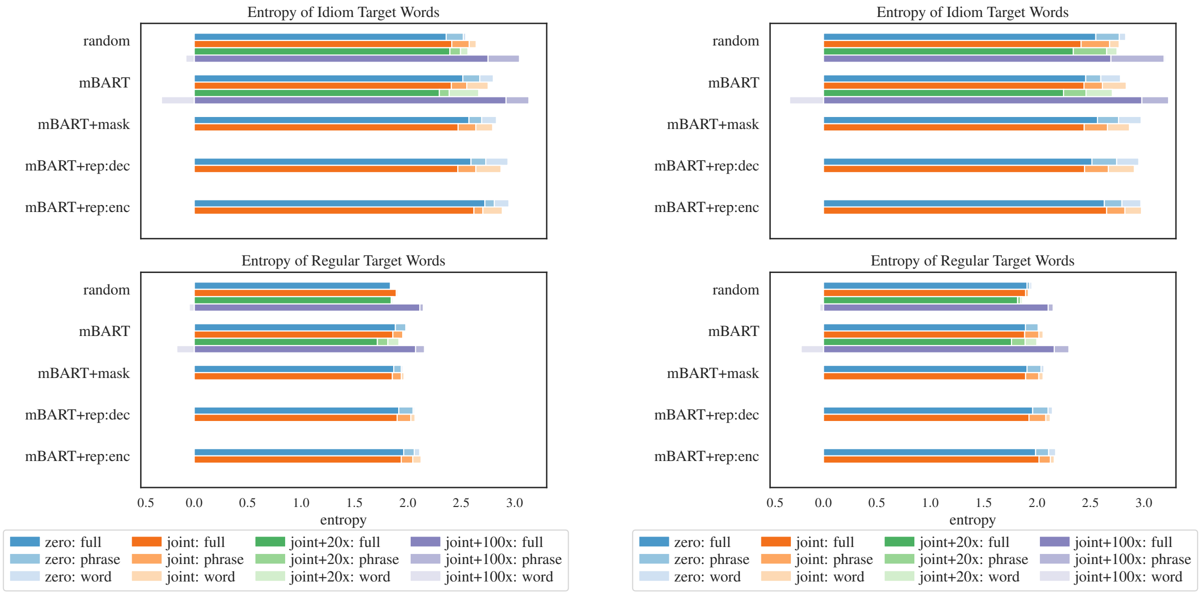
when we encode idioms with word context.

D.2 Variation in Translation Likelihood

With this probe, we test how the variation of the idiom encoder representations affects the likelihood of the reference translations. Figure 20, shows a visual example of how this probe works. Specifically, we translate a sentence pair with teacher-forcing, but we replace the encoder idiom token representations, before passing them to the decoder, with the representations obtained after encoding them with different contexts. Finally, we measure the perplexity of the reference translation under the model. Figure 19, compares models across splits and contexts.

Overall, the results are very consistent across both language pairs and all model variants. We also observe that the behaviour of all the noisy finetuned mBART models very similar behaviour, across all contexts. However, there is a small increase in the perplexity assigned under the "mBART+replace(dec)" variant. This is expected, as this model was trained with decoder dropout, which affect the LM capabilities of the model, and consequently makes it assign an overall smaller probability to all sentences. This increase in perplexity is observed in both language pairs, but is more pronounce in en→fr.

The more we upsample the idiom-train sentence pairs, the less probable other sentences become under the model. Surprisingly, 100x upsampling causes the pretrained model to yield lower perplexity with narrower context (i.e., lighter bars have negative length), "reverting" some of the effects of overfitting. However, we don't have a satisfying explanation for this behaviour[11]. This phenomenon is observed in both language pairs.



(a) Results for the en→fr models.

(b) Results for the en→es models.

Figure 21: Comparison of model uncertainty during the translation of regular-vs-idiom words. The length of each (lighter) bar encodes the difference from its (darker) bar to its left. Negative bar lengths indicate a decrease relative to the (darker) bar before it.

D.3 Decoder Uncertainty

Next, we focus on how the token-level uncertainty of the decoder varies while it translates idiom vs. non-idiom words (Figure 8). For each model, first, we translate each sentence pair with teacher-forcing[12]and then measure the entropy of the decoder's distributions for each target token.

Once more, this probe reveals that the models in both language pairs behave similarly. As mentioned in the main paper, the distributions of words that translation the idiom phrase have significantly larger entropy that the rest. This demonstrates that the models clearly are much more uncertain when translating idioms, even the pretrained ones.

Including and upsampling 20x the idiom-train data is helpful, but extreme upsampling (i.e., 100x) is universally harmful, although we observe a drop in uncertainty with word-level context. This is more pronounced when translating idiom words and suggests that models have overfitted to the words of the idiom phrases.