Robust Open-Vocabulary Translation from Visual Text Representations

Elizabeth Salesky[1] and **David Etter**[2] and **Matt Post**[1,2]
[1]Center for Language and Speech Processing
[2]Human Language Technology Center of Excellence
Johns Hopkins University
Baltimore, Maryland, USA

Abstract

Machine translation models have discrete vocabularies and commonly use subword segmentation techniques to achieve an 'open-vocabulary.' This approach relies on consistent and correct underlying unicode sequences, and makes models susceptible to degradation from common types of noise and variation. Motivated by the robustness of human language processing, we propose the use of *visual text representations*, which dispense with a finite set of text embeddings in favor of continuous vocabularies created by processing visually rendered text. We show that models using visual text representations approach or match performance of text baselines on clean TED datasets. More importantly, models with visual embeddings demonstrate *significant robustness* to varied types of noise, achieving e.g., 25.9 BLEU on a character permuted German–English task where subword models degrade to 1.9.

1 Introduction

Machine translation models degrade quickly in the presence of noise, such as character swaps or misspellings (Belinkov and Bisk, 2018; Khayrallah and Koehn, 2018). These issues can be mitigated with techniques such as normalization, adding synthetic noisy training data (Vaibhav et al., 2019), or simply moving to larger data settings. But, more types of noise and variation exist than can easily be enumerated or normalized, and their combinatorics often present problems even when attempts are made to address them, in addition to requiring careful thought and adding complexity to the model training process. Part of the reason for this brittleness is the reliance of MT systems on *subword segmentation* (Sennrich et al., 2016) as the solution for the open-vocabulary problem, since even minor variations in text can result in very different token sequences, needlessly fragmenting the data.

Phenomena	Word	BPE	
Vowelization	كتاب	كتاب	(1)
	الكِتابُ	الك · ِ · ت · اب · ُ	(5)
Misspelling	language	language	(1)
	langauge	la · ng · au · ge	(4)
Visually Similar Characters	really	really	(1)
	rea11y	re · a · 1 · 1 · y	(5)
Shared Character Components	확인한다	확인 · 한 · 다	(3)
	확인했다	확인 · 했다	(2)

Figure 1: Examples of common behavior which cause divergent representations for subword models. BPE models shown have vocabularies of size 5k.

Humans, in contrast, are remarkably robust to text permutations (Rayner et al., 2006) or visually similar input such as l33tspeak (Perea et al., 2008). It stands to reason that one source of this robustness is that humans process text, not from discrete unicode representations, but *visually*, and that providing models access to this kind of representation might yield more human-like robustness.

Drawing on this, we propose to use visual text representations of translation input. Our translation models still consume text, but instead of creating an embedding matrix from subwords, we render the text as images, divide the image into a sequence of overlapping slices, and produce representations using techniques from optical character recognition (OCR). The rest of the translation architecture remains unchanged. These models therefore contain both visual and distributional information about the input, which may allow them to learn robust and generalizable input representations even in the presence of various kinds of noise.

After presenting the visual text embedder (Section 2), we report results on small-data scenarios

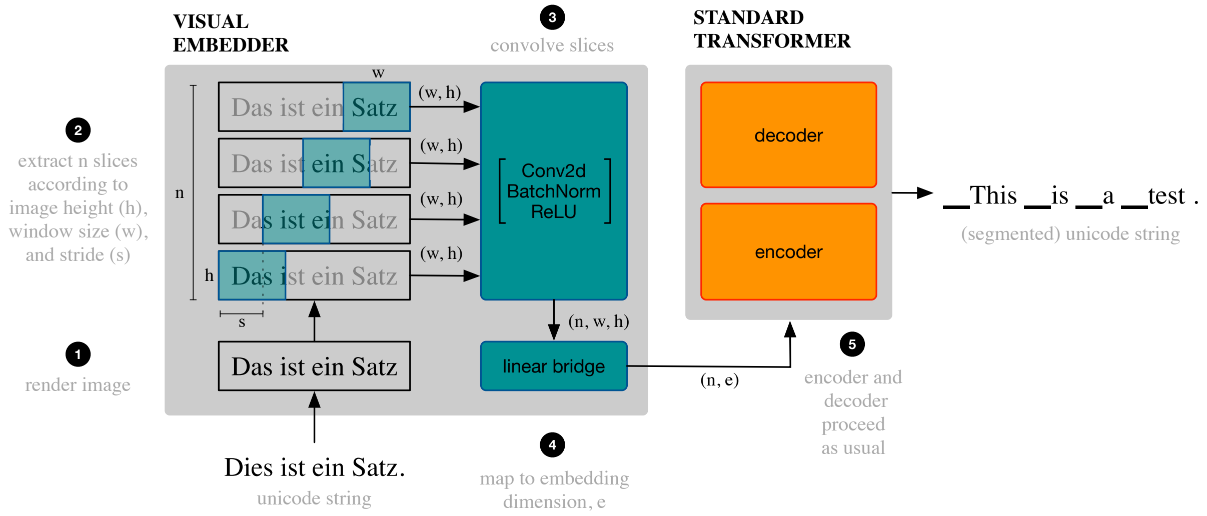


Figure 2: Visual text architecture combines network components from OCR and NMT, trained end-to-end.

across seven language pairs with several scripts. We show that translation models using visual text representations are able to match or approach the performance of subword models (Section 4), especially in languages using Latin script. We then look at a variety of noisy data scenarios, showing that visual text models exhibit a remarkable robustness to induced noise (Section 5).

To summarize, we:

- propose the use of visual representations, motivated by human text processing, as a means for achieving robustness and simplifying data preprocessing;
- demonstrate the potential of visual representations for machine translation across a range of languages and scripts; and
- show significant improvements to model robustness to synthetic and natural noise.

2 Visual Text Embedder

2.1 Rendering text as images

To create visual text representations, we first render text as images. Each sentence is rendered from raw text as a single image in grayscale with 1 channel rather than 3 color channels; no subword processing is used at all. The image height h is a function of the maximum height of the characters given the font and font size, while the image width w is variable based on the font and sentence length. We extract slices using sliding windows, similar to feature extraction for speech processing. Each window is of a specified length w and full height h, extracted at intervals s determined by a set stride (see Figure 2). These slices are analogous to text tokens, and can be batched together and processed in parallel. We experimentally tune each of these parameters per language pair, discussed in Section 3.3.

2.2 Architecture

Our visual text model replaces only the source embedding matrix of a standard text translation model, shown in Figure 2. Embeddings typically refer to entries in a fixed size weight matrix. In place of source embeddings, we use the continuous outputs of convolutional blocks run over rendered text as the input to a Transformer (Vaswani et al., 2017). The full model is trained end-to-end with the typical cross-entropy objective.

While OCR models for tasks such as handwriting recognition require depth that impacts training and inference speed, our task differs significantly. OCR tasks contend with varied image backgrounds, varied horizontal spacing, and varied character 'fonts,' sizes, colors, and saliency. Visually rendered text is uniform along each of these characteristics by construction. Accordingly, we can use simpler image processing and model architectures without impact to performance.

Our experiments use a single convolutional block ($c = 1$) followed by a linear projection to produce flattened 1D representations as used by typical text-to-text Transformer models, but here the 'embeddings' represent a continuous vocabulary rather than a discrete embedding matrix. A convolutional block comprises three pieces: a 2D convolution, followed by 2D batch normalization, and a ReLU layer. The 2D convolution is done with padding of 1, a kernel size of 3, a stride of 1, and an output channel size of 1. These settings result in no change in dimensions between the block inputs and outputs. By contrast, VistaOCR (Rawls et al., 2017) uses $c = 7$ convolutional blocks which iteratively grow the channel axis from an initial 3-color channels to 256, with 2 additional interleaved max-pooling layers. When $c = 0$, the model is akin to the Vision Transformer (Dosovitskiy et al., 2021) from image classification where attentional layers are applied directly to image slices[1] after a flattening linear transformation.

The subsequent Transformer follows the architecture of our text translation models (Section 3.2). All models are trained using a modified version of **fairseq** (Ott et al., 2019). Code and config files to replicate our experiments will be released upon publication.

3 Experimental Setup

3.1 Datasets

MTTT. We use the Multitarget TED Talks Task (MTTT), a collection of TED talks datasets with ∼200k training sentences and multi-parallel dev and test sets (Duh, 2018), to compare visual text and text translation models across multiple source languages and scripts into English. Specifically, we use the data for the Arabic (ar), Chinese (zh), Japanese (ja), Korean (ko), Russian (ru), French (fr), and German (de) to English (en) tasks.

MTNT. To evaluate model robustness on data with naturally occurring noise, we use the Machine Translation of Noisy Text (MTNT) test sets (Michel and Neubig, 2018). The MTNT test sets used were created from comments from Reddit in French, German, and Japanese which have been professionally translated from English. By virtue of their domain, these test sets contain "noisy" text with natural typos, semantic use of visually similar characters, abbreviations, grammatical errors, emojis, and more. MTNT has recently been used for evaluation in the WMT'19 and '20 Robustness tasks (Li et al., 2019; Specia et al., 2020).

WIPO. We additionally use the World Intellectual Property Organization (WIPO) COPPA-V2 corpus (Junczys-Dowmunt et al., 2016) to evaluate robustness on data with naturally occurring noise for Russian-English. The WIPO corpus consists of parallel sentences from international patent application abstracts.

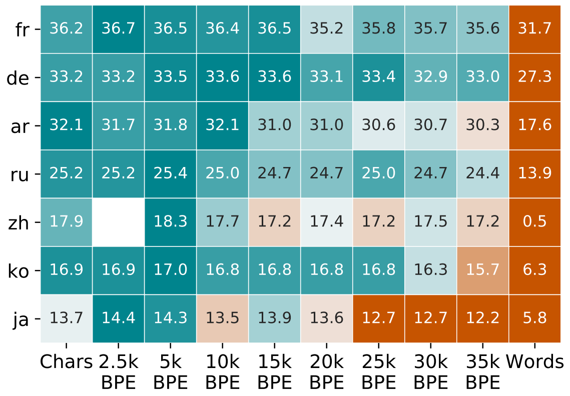


Figure 3: Baseline results on MTTT TED across BPE segmentations with optimized batch size.

3.2 Text models

All baseline text models are trained using **fairseq** (Ott et al., 2019). For the 7 language pairs selected from the MTTT TED dataset, we follow the recommended **fairseq** architecture and optimization parameters for IWSLT'14 de-en which is of the same size and domain – 6 layers each for encoder and decoder, with 4 attention heads per layer, with slight modifications to batch size, vocabulary, and label smoothing $p = 0.2$.

We tune the subword vocabulary independently for each language pair and dataset. We did not see a difference in performance with joint rather than separate vocabularies, so use separate vocabularies with the target vocabulary held constant to provide a more direct comparison between our unicode text baselines and visual text models. We tuned ∼5k BPE intervals from 2.5k–35k[2] to optimize source language BPE granularity with the target (English) vocabulary constant at 10k BPE. We additionally compare character-level and word-level models; to produce word-level segmentations for Chinese, we use **jieba**,[3] and for Japanese, we use **kytea** (Neubig et al., 2011). The character vocabulary for Chinese is greater than 2.5k so we do not have a BPE model of this size. Our best performing BPE models used source vocabularies of approximately 5k (see Figure 3).

[1] Our model also differs in that it uses overlapping slices extracted only along the original image width.

[2] For these datasets, ∼40k BPE recovers words.

[3] https://github.com/fxsjy/jieba

	Text		Visual text			
Lang	BPE	char	$s = 5$	$s = 10$	$s = 15$	$s = 20$
ar	24.4	78.9	97.1	48.8	32.7	24.6
de	32.3	104.3	130.5	65.5	43.8	33.0
fr	28.8	107.6	130.2	65.4	43.7	32.9
ja	22.5	36.9	95.5	48.0	32.1	24.2
ko	24.7	50.8	97.0	48.7	32.6	24.6
ru	27.1	94.7	132.7	66.6	44.5	33.5
zh	23.0	29.8	75.6	38.1	25.5	19.3
Time	1.0	2.3	3.9	2.0	1.4	1.2

Table 1: Average sequence lengths over the training data varying the stride, s. The bottom row denotes the training time multiplier for 100 epochs, relative to the best text model (BPE).

		Text	Visual Text	
			$c = 7$	$c = 1$
ar	Arabic–English	32.1	30.2	30.0
de	German–English	33.6	34.3	33.6
fr	French–English	36.7	35.3	35.5
ja	Japanese–English	14.4	12.8	11.5
ko	Korean–English	17.0	16.2	15.2
ru	Russian–English	25.4	23.3	23.8
zh	Chinese–English	18.3	17.0	14.4

Table 2: Translation results in BLEU on MTTT TED show our best visual text models approach parity with our best text baseline models. n = number of convolutional blocks.

We jointly tuned batch size and subword segmentation for each language pair and found significant (1-15 BLEU) improvements with a larger batch of 16k tokens over the recommended 4096. For Chinese, Korean, and Japanese specifically, improvements averaged 12-15 BLEU within each subword granularity by increasing batch size to 8k or higher. Our baselines improve ∼2 BLEU over previous work on the MTTT dataset (Shapiro and Duh, 2018).

3.3 Visual text models

To visually render text, we use the `pygame` Python package[4] with the Google Noto font family.[5] For Latin and Cyrillic scripts, we use NotoSans-Regular; for Arabic, NotoNaskhArabic-Regular; and for the ideographic languages, NotoSansCJKjp-Regular. All fonts are rendered at 8 points, which produces images with a height of 23 pixels.

Our visual text models combine the visual text embedder from Section 2 with the exact Transformer architecture described above for text translation models. We experiment with window size and stride in the following section.

While our visual text models remove the source embedding matrix, they add parameters from convolution blocks used to compute the representations, increasing overall model parameters from 36.7M to about 36.9M, varying slightly, depending on the window size. Overlapping sliding windows result in longer source sequences than text baselines, and a consequent increase in training time (Table 1). Rendering text is negligible at inference time, comparable to subword segmentation, and can be done as part of preprocessing for training.

4 Chasing translation parity

State-of-the-art translation models use distributional embeddings for subwords, which are representative of the vocabulary distribution of a given language's (training) text and enable models to avoid the "OOV problem." Subword granularity is a parameter to be tuned for each language pair and task (Ding et al., 2019; Salesky et al., 2020); with our sliding window and convolutional visual representations, we have the opportunity to avoid hard decisions about subword granularity which create a finite model vocabulary. We first ask if we can recover the best results from models with optimized subword segmentations using our visual text representations, rather than typical distributional embeddings learned for the translation task.

Table 2 compares our visual text models to our best text baselines on MTTT. We see that we can nearly recover the best results from the most optimal BPE segmentation *without* explicit input segmentation, solely from visual representations with a sliding window. While added visual capacity through a greater number of convolutional blocks ($c = 7$) can improve translation results, this depth comes at a cost: 2.6M additional parameters and a 5× decrease in training speed compared to $c = 1$. Our analysis focuses on the $c = 1$ case.

We experiment with window size and stride for visual text experiments. Window length is always greater than stride length in order that no text is dropped. We see similar overall trends across

[4] https://www.pygame.org

[5] https://www.google.com/get/noto

DE-EN	$c = 1$			
window↓/stride→	10	15	20	25
15	33.4	–		
20	33.1	32.9	–	
25	33.3	31.3	30.3	–
30	33.6	32.9	32.4	30.9

Table 3: German–English BLEU scores on MTTT, tuning stride and window length with fixed batch size.

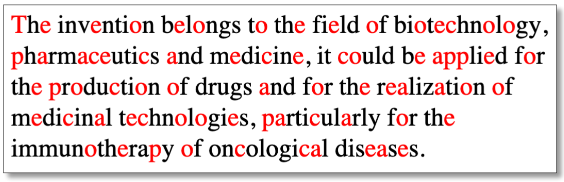

The invention belongs to the field of biotechnology, pharmaceutics and medicine, it could be applied for the production of drugs and for the realization of medicinal technologies, particularly for the immunotherapy of oncological diseases.

Figure 4: Different unicode codepoints may render similarly. In this English example from WIPO (Junczys-Dowmunt et al., 2016), all characters in red are not from the Roman alphabet but Cyrillic.

language pairs. Table 3 shows varied window length and stride values for de-en, with a batch size of 10k and font size 8: additional language pairs can be found in Appendix A. We do not observe a consistent pattern across different window sizes, but as stride length increases (creating less overlap between image tokens) there is a consistent performance drop off. Our best results use stride 10 or 15.

5 Robustness to Noise

We hypothesize that without a fixed vocabulary and with associations between visually similar character spans, our visual text models will be more robust to noise than text-based representations, where noise causes divergent subword representations (see Figure 1 for motivating examples). To test this, we evaluate on two different settings: induced synthetic noise, and naturally occurring noise from sources such as Reddit. Synthetic noise allows us to test various settings for all language pairs, while natural noise is limited by dataset availability.

5.1 Synthetic noise

Inducing noise enables us to control the type and frequency with which noise occurs. We compare two types of synthetic noise: visually similar characters (e.g., l33tspeak, unicode codepoints which render visually similar) and character permutations (e.g., Cmabrigde). For all synthetic noise experiments, we induce noise at the token-level on the source side of our baseline dataset, MTTT TED. Each token may be replaced with probability p from $p = 0.1$ to $p = 1.0$ by intervals of 0.1. An example of each synthetic noise type and model outputs can be found in Table 4.

Visually similar characters. Different unicode characters may share visually similar characteristics. Such characters may be substituted intentionally, such as in l33tspeak where characters such as numbers are used in place of visually similar Roman alphabet letters, or unintentionally, where characters from another script appear in place of the expected unicode codepoints for a given language and script due to e.g., use of multiple keyboards or OCR errors (Rijhwani et al., 2020). For some languages without a unicode standard, multiple unicode sequences which render the same are all in common use (e.g., Pashto). As shown in Figure 4, such errors can be very inconspicuous.

We induce visual noise in the form of visually similar Latin characters in place of Cyrillic characters for Russian (**`unicode`**), and in the form of `l33tspeak` for French and German, which use the Latin alphabet. Figure 5 shows that the visual text model has almost no degradation in performance with `unicode` noise, even when 100% of characters with a mapping to another visually similar unicode codepoint have been substituted. However, the text model quickly degrades as substitutions cause mismatches with BPE vocabularies. Character-based models are also unable to handle OOV codepoints, and characters in extremely novel context as found with visual noise:

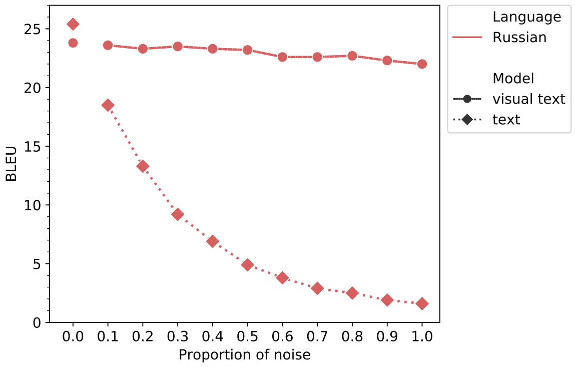


Figure 5: **Visual noise: `unicode`**. Inducing visually similar unicode codepoint differences for Russian does not affect visual text, but breaks BPE representations.

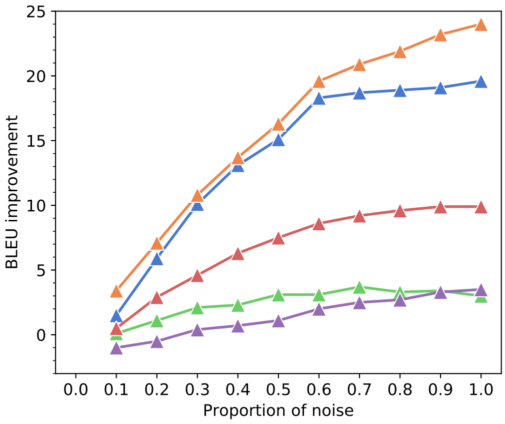


Figure 6: **Character permutations: `swap`**.

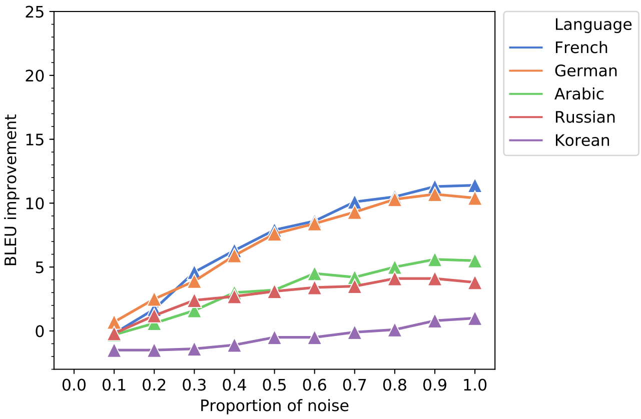


Figure 7: **Character permutations: `cam`**.

at $p = 0.5$, our character model has a disappointing 0.2 BLEU.

While different unicode codepoints may be visually near-identical and as such our visual text models are able to neatly solve this problem, other such visual noise is not as directly matched. In the case of **`l33tspeak`**, readers understand from the unexpected presence of a number that a substitution has been made and are able to form a mapping to a similar alphabetic letter. However, '4' and 'a' are not necessarily more visually similar in many fonts than say '7' and 'z'; conventional use often dictates substitutions. Figure 8 shows that while both visual text models and text models are negatively affected by induced **`l33tspeak`**, the visual text models for both language pairs significantly outperform the text models in these conditions. With up to 30% of tokens containing **`l33tspeak`** mappings for all available characters, the visual text models for both German and French perform >5 BLEU better than the text baselines.

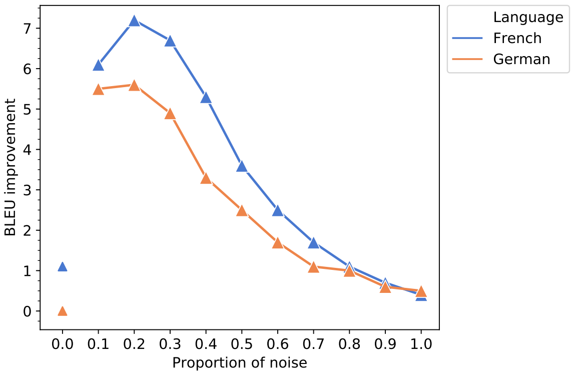


Figure 8: **Visual noise: `l33tspeak`**. Improvements with visual text diminish with higher levels noise.

While it is easy to envision a pipeline using OCR could solve the unicode noise problem, a perfect OCR model would return the **`l33tspeak`** mappings, leaving this task the same as with a text translation model.

Character permutations are challenging both for subword-based translation models, which necessarily back off to smaller units in the presence of OOVs (see Figure 1), as well as for character-based models (Belinkov and Bisk, 2018) unless word-internal order is modeled (Sakaguchi et al., 2017). Here we experiment with two types of synthetic noise used by Belinkov and Bisk to compare visual text models to text baselines.

`Swap` : Swapping adjacent characters (e.g. *language*→*langauge*) is common when typing quickly. We perform one swap per word. This noise is applied to words of length ≥2.

`Cam` : The purported Cambridge spelling experiment of spam mail fame illustrates the remarkable robustness of humans to character permutations,[6] when the first and last character are unchanged (e.g. *language*→*lnagauge*). To enable word-medial permutations, this noise is applied to words of length ≥4.

We do not apply character permutations to Chinese or Japanese–English tasks, as the majority of tokens contain two or fewer characters after word segmentation.

Visual text representations result in significant improvements for character permutations, particularly at higher levels of noise. Figure 6 and Figure 7 shows the stark contrast in relative performance between the two models: though a slight

[6] with a cost to reading speed (McCusker et al., 1981; Rayner et al., 2006).

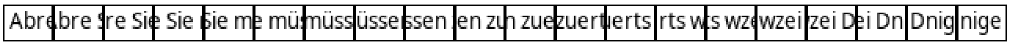
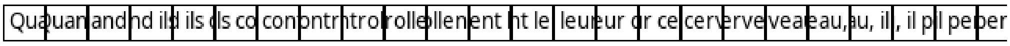
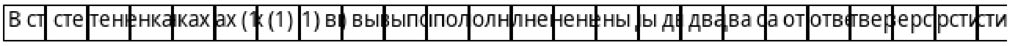
German–English	
src	Aber Sie müssen zuerst zwei Dinge über mich wissen.
`swap` 0.5	Abre Sie müssen zuerts wzei Dnige über mcih wisse.n
in$_{\text{vis}}$	 . . .
out$_{\text{vis}}$	But you first have to know two things about me.
in$_{\text{text}}$	_Ab re _Sie _müssen _zu ert s _w z ei _D n ige _über _m ci h _ wiss e . n
out$_{\text{text}}$	Chris Asdrneno: Nein, du kannst mit den deri Mineutn noch nhict anfgnean.
ref	But first you need to know two things about me.
French–English	
src	Quand ils controllent leur cerveau, il peuvent controller leur douleur.
`cam` 0.2	Quand ils controllent leur cerveau, il penevut clntreloor leur douleur.
in$_{\text{vis}}$	 . . .
out$_{\text{vis}}$	When they control their brain, it's picking up their pain.
BPE	_Quand _ils _contr ol l ent _leur _cerveau , _il _p en e v ut _c l nt re lo or _leur _douleur .
out$_t$	And she led me to an orthopedic surgeon, using free.
ref	When they control their brain, they can control their pain.
Russian–English	
src	В стенках (1) выполнены два отверстия (7), (8).
in$_{\text{vis}}$	 . . .
out$_{\text{vis}}$	There are two holes in the wall.
in$_{\text{text}}$	_В _стен как _(1) _выполнен ы _два _от вер сти я _(7) , _(8) .
out$_{\text{text}}$	In the walls (1) there are provided two openings (7) and (8).
ref	(1) There are two holes in the wall.

Table 4: Examples of noisified text and the respective inputs and outputs of the baseline text and the visual systems. The Russian example comes from WIPO dataset, and the German and French ones from MTTT.

gap in performance remains for some of our models on clean text, with even 10% induced noise this gap has been closed. Improvements of up to 24 BLEU on German–English concretely mean that our visual text model achieves 25.9 BLEU on a task where the subword-based model has degraded to 1.9 BLEU. Figure 9 in Appendix B shows absolute degradation in performance BLEU for each individual model and permutation type.

Character permutations exhibit the opposite trend of visual noise: while improvements over text models decreased when more tokens contained visual noise, for permutations improvements strongly increased with greater levels of noise. This may be because visual noise involves character *substitutions* rather than *permutations*. Permutations affect a greater percentage of the character sequence for a given token, which shatter BPE representations. While BPE models can use context to recover when only 10% of tokens contain permutations, at higher levels they cannot. When 100% of tokens contain **swaps**, for example, the German 5k BPE model backs off to 2.25× more subwords (most word types become character sequences) to represent test sentences than with unnoised text.

The visual text image slices contain the same permuted character sequences as the raw text: however, rather than an embedding which represents distributional properties of a subword (as learned for translation), the input to the visual text Transformer is a vector which represents the utility of visual features across a character span (for translation). Though the model is not trained on perturbations, we hypothesize that use of representations computed across multiple characters in each slice results in a model with greater robustness to perturbations.

5.2 Natural noise

Natural noise as found in data such as Reddit contains many additional types of noise to those we induced above, including keyboard typos (where nearby keys are substiyuted), substitutions of phonetically similar characterz or worts, unconventional s p a c e s and repetitionsss for effect or error, natural mispelling, and noisy spans which extend beyond individual tokens, among others. Parallel text created from 'found' data (MTNT: Reddit; WIPO: patents) contains such noise in natural contexts.

	MTNT			WIPO
Model	**fr-en**	**ja-en**	**de-en**	**ru-en**
Text, subword	26.4	4.3	18.2	9.9
Text, character	26.7	3.7	20.7	10.3
Visual text	26.2	3.6	20.4	10.5

Table 5: Translation results on test sets with naturally occurring noise.

Table 5 compares visual text models to text models using both subword and character-level representations on MTNT and WIPO test sets. We test our models in a zero-shot setting, without continued training for adaptation. The domain mismatch of these test sets provides a challenge for all models. We see that character-level models are in some cases more robust than subwords, but are unable to recover from the variation in others (ja-en). The visual text models perform competitively with character-level models for German–English, where we have reached parity on our clean data case (Table 2), and Russian–English, where the WIPO patent data has a significant number of unicode OCR errors as illustrated in Figure 4 and also occasional Roman alphabet characters for e.g., chemical formulas: 3% of characters in the Russian WIPO test set source come from outside the Cyrillic unicode codepoint range.

6 Related Work

A common point of interaction between vision and natural language is in multimodal image captioning with language grounding (Vinyals et al., 2015; Xu et al., 2015), where the images contain objects or scenes rather than text, and optical character recognition (Shi et al., 2017; Rawls et al., 2017).

Visual representations of text have previously been explored for other NLP tasks, primarily for Chinese characters. Liu et al. (2017) used visual representations from CNNs over rendered text in Chinese, Japanese, and Korean for text classification. Dai and Cai (2017) similarly used convolutions over character-level images for Chinese for downstream language modeling and word segmentation. Sun et al. (2019) created dense square renderings of text to use convolutions for both Chinese and Latin alphabets for downstream sentiment analysis.

In machine translation, an initial idea was to use linearized bitmaps of Chinese characters to initialize word embeddings for early pre-Transformer seq2seq models (Aldón Mínguez et al., 2016; Costa-jussà et al., 2017). More recently, Mansimov et al. (2020) explored image-to-image translation; in this setting, both source and target language representations are derived from images only, not text. They were motivated in part by a desire to do away with fixed, pre-defined segmentation models and even vocabularies—an advantage that our approach shares. Our approach is quite different. It is image-to-text, instead of image-to-image; we produce visual representations from fixed-size image slices, whereas they (appear to) render a single compressed sentence encoding; and we predict text, instead of pixels. Their work is exploratory, and yields low results.

Previous work has explored the impact of synthetic and natural noise on neural MT (Belinkov and Bisk, 2018), and the use of character-aware word embeddings (Kim et al., 2016; Sakaguchi et al., 2017; Cherry et al., 2018; Clark et al., 2021) to increase generalizability and robustness. While research regularization and dropout techniques for BPE (Kudo, 2018; Provilkov et al., 2020) have improved model robustness, discrete vocabulary sets still creates challenge in many use cases. Recent work has also explored byte-level BPE (Radford et al., 2019; Wang et al., 2020) to create models which are not restricted to the unicode ranges seen in training, though models using BBPE often require additional training examples.

7 Conclusion

We introduced visually rendered text for continuous open-vocabulary translation. We showed that our models, trained in the low-resource TED setting on seven language pairs, approach or match the performance of text-based representations. Further, we showed that visual text models are more robust to many kinds of induced noise—including, but not limited to, visually similar characters.

We believe this approach holds a lot of promise. The experiments here barely begin to explore the potential of this approach for machine translation alone. For next steps, we think it is important to conduct a deep investigation into the visual text architecture and parameter optimization, and to extend our experiments to larger-data scenarios. Since our approach does away with text-based segmentation and discrete vocabularies, visual text models could be applied to new languages and scripts without requiring transliteration or normalization. Following this, we believe this representation approach could be successful for other NLP tasks, such as language ID (Caswell et al., 2020, Table 2).

Acknowledgments

We thank Adi Renduchintala for early enthusiasm and proofs of concept, Paul McNamee for data pointers and examples (particularly Figure 4), and Facebook for a 2019 research award to the third author.

References

David Aldón Mínguez, Marta Ruiz Costa-Jussà, and José Adrián Rodríguez Fonollosa. 2016. Neural machine translation using bitmap fonts. In *Proceedings of the EAMT 2016 Fifth Workshop on Hybrid Approaches to Translation (HyTra)*, pages 1–9.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language ID in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. Canine: Pre-training an efficient tokenization-free encoder for language representation. *arXiv preprint arXiv:2103.06874*.

Marta R. Costa-jussà, David Aldón, and José A. R. Fonollosa. 2017. Chinese–spanish neural machine translation enhanced with character and word bitmap fonts. *Machine Translation*, 31(1):35–47.

Falcon Dai and Zheng Cai. 2017. Glyph-aware embedding of chinese characters. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 64–69.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Kevin Duh. 2018. The multitarget TED talks task. http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/.

Marcin Junczys-Dowmunt, Bruno Pouliquen, and Christophe Mazenc. 2016. Coppa v2. 0: Corpus of parallel patent applications building large parallel corpora with gnu make. In *4th Workshop on Challenges in the Management of Large Corpora Workshop Programme*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Pino, and Hassan Sajjad. 2019. Findings of the first shared task on machine translation robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 91–102, Florence, Italy. Association for Computational Linguistics.

Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig. 2017. Learning character-level compositionality with visual features. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2059–2068.

Elman Mansimov, Mitchell Stern, Mia Xu Chen, Orhan Firat, Jakob Uszkoreit, and Puneet Jain. 2020. Towards end-to-end in-image neural machine translation. In *Proceedings of the First International Workshop on Natural Language Processing Beyond Text*, pages 70–74.

Leo X McCusker, Philip B Gough, and Randolph G Bias. 1981. Word recognition inside out and outside in. *Journal of Experimental Psychology: Human Perception and Performance*, 7(3):538.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Manuel Perea, J. A. Duñabeitia, and M. Carreiras. 2008. R34d1ng w0rd5 w1th numb3r5. *Journal of experimental psychology. Human perception and performance*, 34 1:237–41.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. Bpe-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Stephen Rawls, Huaigu Cao, Senthil Kumar, and Prem Natarjan. 2017. Combining convolutional neural networks and LSTMs for segmentation free OCR. In *Proc. ICDAR*.

K. Rayner, S. White, Rebecca Lynn Johnson, and S. Liversedge. 2006. Raeding wrods with jubmled lettres. *Psychological Science*, 17:192 – 193.

Shruti Rijhwani, Antonios Anastasopoulos, and Graham Neubig. 2020. OCR Post Correction for Endangered Language Texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5931–5942, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. 2020. Optimizing segmentation granularity for neural machine translation. *Machine Translation*, 34(1):41–59.

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

Pamela Shapiro and Kevin Duh. 2018. Bpe and charcnns for translation of morphology: A cross-lingual comparison and analysis. *arXiv preprint arXiv:1809.01301*.

Baoguang Shi, X. Bai, and Cong Yao. 2017. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2298–2304.

Lucia Specia, Zhenhao Li, Juan Pino, Vishrav Chaudhary, Francisco Guzmán, Paul Michel, Graham Neubig, Hassan Sajjad, Nadir Durrani, Yonatan Belinkov, Philipp Koehn, and Xian Li. 2020. Findings of the wmt 2020 shared task on machine translation robustness. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Baohua Sun, Lin Yang, Catherine Chi, Wenhan Zhang, and Michael Lin. 2019. Squared english word: A method of generating glyph to use super characters for sentiment analysis. In *AffCon@ AAAI*.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. 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 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. 2015. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9154–9160.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

A Parameter tuning

Results of tuning window length and stride by language pair for MTTT; results for de-en can be found in the main text (Table 3). Window length is always greater than stride length in order that no image slices are dropped. With $c = 1$ convolutional blocks, we observe occasional instability that we do not see with $c = 7$, which we suspect additional experimentation with batch size and model regularization may stabilize.

FR–EN	$c = 1$			
window↓/stride→	10	15	20	25
15	33.9	–	■	■
20	35.1	34.5	–	■
25	35.5	34.8	34.0	–
30	35.0	33.5	33.5	32.6

ZH–EN	$c = 1$			
window↓/stride→	10	15	20	25
15	0.5	–	■	■
20	14.4	13.4	–	■
25	0.5	0.6	0.5	–
30	0.5	0.6	4.3	5.3

AR–EN	$c = 1$			
window↓/stride→	10	15	20	25
15	29.5	–	■	■
20	29.7	28.5	–	■
25	27.1	24.0	11.5	–
30	30.0	28.6	26.9	25.4

KO-EN	$c = 1$			
window↓/stride→	10	15	20	25
15	15.2	–	■	■
20	14.8	14.9	–	■
25	14.7	14.3	14.1	–
30	14.6	14.3	13.3	12.3

RU–EN	$c = 1$			
window↓/stride→	10	15	20	25
15	19.6	–	■	■
20	0.6	0.5	–	■
25	23.8	22.8	22.3	–
30	23.3	23.2	22.5	21.4

JA–EN	$c = 1$			
window↓/stride→	10	15	20	25
15	10.7	–	■	■
20	11.5	10.5	–	■
25	10.9	10.9	8.2	–
30	11.1	10.0	8.9	8.0

Table 6: Translation results for MTTT, tuning stride and window length with fixed batch size $10k$ and font size 8.

B Additional robustness results

Here we show character permutation results isolated by model and noise type. Each plot shows the degradation in performance of a given model with different proportions of induced noise, relative to the performance of the same model on the uncorrupted text. As more noise is added, the visual text models degrade at significantly lower pace.

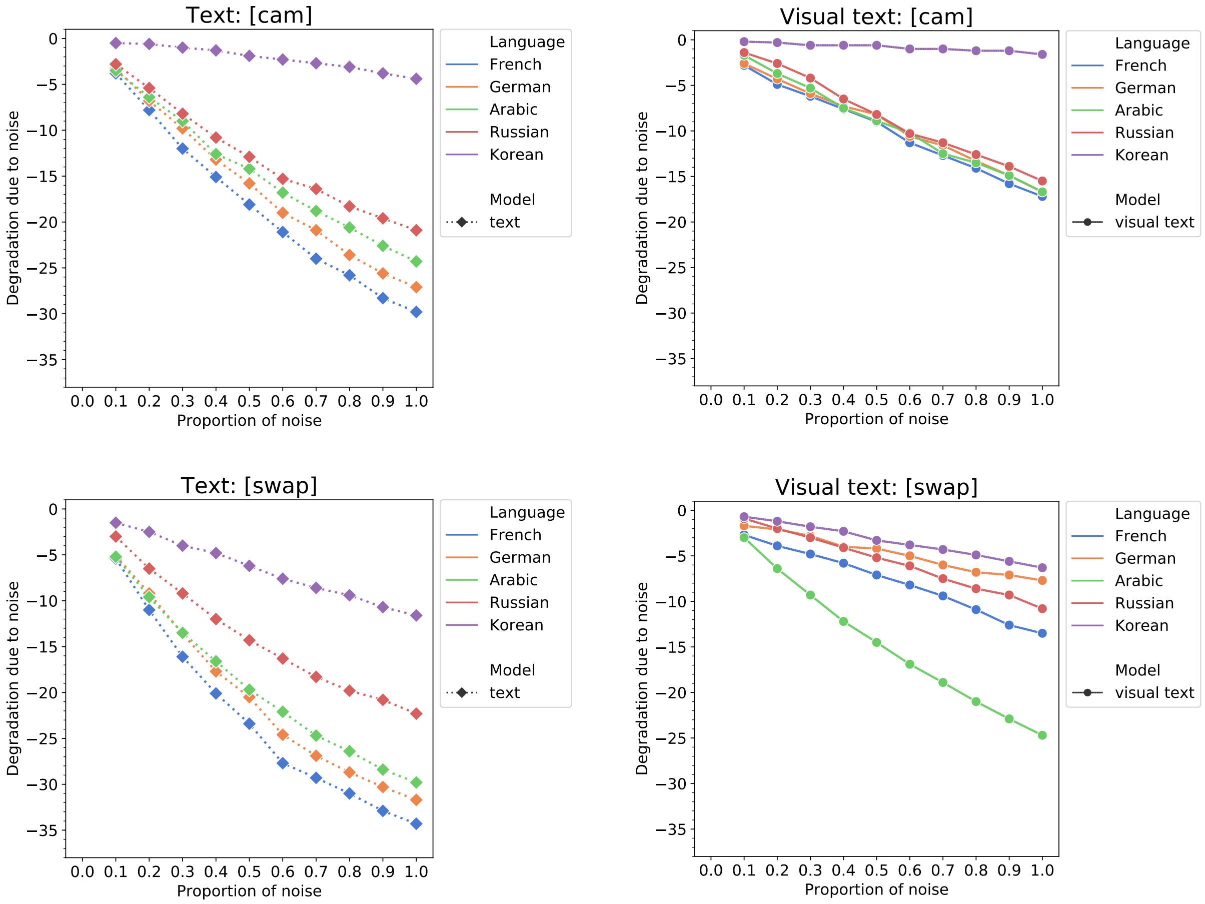


Figure 9: **Degradation due to noise in the form of character permutations**. Each point plots how much worse the model does at a given amount of noise, relative to the same model on uncorrupted text. [Top] **`cmabirdge`** word-internal permutations; [Bottom] **`swap`** of two characters within a token. [Left] text model baselines; [Right] visual text models.