An Exploration of Placeholding in Neural Machine Translation

Matt Post[†◇] **Shuoyang Ding**[†] **Marianna J. Martindale**[‡] **Winston Wu**[†]
† Center for Language and Speech Processing, Johns Hopkins University
‡ iSchool, University of Maryland College Park
◇ Human Language Technology Center of Excellence, Johns Hopkins University
post@cs.jhu.edu, {dings, wswu}@jhu.edu
mmartind@umiacs.umd.edu

Abstract

Phrase-based machine translation provides the system developer with controls that enable fine-grained control over machine translation output. One approach to provide similar control in neural machine translation is *placeholding* (herein called *masking*), which replaces input tokens with masks which are replaced with the original input text in post-processing. But is this a good idea? We undertake an exploration of masking in French–English and Japanese–English using Transformer architectures. We attempt to quantify whether (and where) masking is necessary with analysis of a baseline system, and then explore numerous parameterization of masking, including post-processing techniques for replacing the masks. Our analysis shows this to be a thorny matter; masks solve some problems but are not perfectly translated themselves.

1 Introduction

Neural machine translation generally produces higher quality output than phrase-based machine translation, especially in high-resource training settings and on in-domain data. However, this improvement has come at the expense of a certain loss of control over how words get translated, since there is no longer a direct link between source words, their translation options, and the ordered decoder output. While nearly everyone has considered this trade to be worthwhile, there lingers a concern about the stability and dependency of NMT performance. Input words are not all equally important, and there are many settings where one would be willing to sacrifice translation quality for a *translation guarantee* that certain input tokens be translated with perfect recall. Common examples include prices on a product page, names and places in a news article, or contact and location information, and other data types, such as URLs.

src	En 2017 Bernard Arnault a gagné...
mask	En NUM NAME NAME a gagné...
out	In NUM NAME NAME a gagné...
align	In NUM$_1$ NAME$_1$ NAME$_2$ a gagné...
detok	In 2017 Bernard Arnault won...

Figure 1: A translation pipeline with masking (placeholding). The indexes denote a permutation of each mask type, and may or may not be an explicit part of the tag.

One attempt to providing these guarantees is the use of placeholders (or *masks*, the term we will use in this paper), where input tokens in a category are replaced by a masked label token (Figure 1). These are then passed through to the output and replaced with the correct translation in post-processing. This ostensibly guarantees that the input term (or its preferred translation) will correctly appear in the output, while at the same time restoring a capability that was easily handled in the old phrase-based paradigm. At the same time, doing so reflects a lack of confidence in the decoder to get this right. This approach has not received much attention in the research literature.

In this paper, we look at this topic in more detail. We focus our attention on *copy* or *pass-through* tokens, which is to say, input tokens that are not translated, but which are simply copied to the output sentence. This includes many different token types that can be recognized by regular expressions (numbers, URLs, email addresses, and Emojis), as well as types for which we can provide a dictionary.

We ask the following questions:

- Are translation guarantees necessary for these types?
- How effective is masking at producing these guarantees?

We experiment in both high resource (FR→EN) and low-resource (JA→EN) language settings.

2 Related Work

The first application of hard masking in neural machine translation was in Luong et al. (2015) and Long et al. (2016), which address the translation of rare words and technical items, but the approach was largely abandoned when sub-word methods (Sennrich et al., 2016) obviated the need. Most similar to this work in spirit is Crego et al. (2016), who mentioned that masking could be used to translate many "pass-through items" but did not conduct any further analysis towards the problem or the solution.

Another solution for handling pass-through items is to add them as constraints during beam search. A number of approaches introduced modifications to beam search that ensured that desired words would be included in the output (Hokamp and Liu, 2017; Chatterjee et al., 2017; Anderson et al., 2017). One problem with these solutions is that decoding time generally grows very quickly with the number of constraints added. Hasler et al. (2018) showed that even two constraints cause decoding speed to increase by as much as five times. Post and Vilar (2018) introduced a fixed-beam-size variant which is constant in the number of constraints, but the constant overhead is still quite high.

In terms of specific token types, Li et al. (2018), Ugawa et al. (2018) and Grundkiewicz and Heafield (2018) studied NMT models with better handling of named entities, either by adding named entity tags or employing transliteration models. Gotti et al. (2014) analyzed how hashtags are translated in the Canadian government tweet corpus and used insights from the analysis to improve their tweet-oriented machine translation system. Radford et al. (2016) conducted corpus analysis on the alignment between natural language text with Emojis.

match type	**examples**
template (regex)	numbers, emoji, URLs, email addresses
direct (dict)	names, cities, states, locations

Table 1: Pass-through candidates can be identified at the class level (via regular expressions) or type level (via direct match against a provided dictionary).

3 Masking

Masking is the context-free replacement of a class of input tokens with a single mask token. The idea is to collapse collections of distributionally similar tokens into a single token that the decoder can then be trained to reliably translate.

Because there has been little formal study of these items, there is no consensus on what should be masked (i.e., what the set of pass-through items is). For this work, the set of items to be masked comes from two different sources (Table 1):

- *Template matches*. This refers to sets of items that can be identified by regular expression. We work with numbers, URLs, email addresses, and emoji (a term we use in a general sense to denote extended non-alphabetic character sets).
- *Dictionary matches*. Tokens or sequences of tokens that are always translated the same way. A canonical example is named entities. These are often identified via dictionary lookup.

Dictionary matches typically contain items that are in fact translated, but we focus on the subset of word tokens that are instead passed through.

3.1 Demasking

At inference time, the masked tokens in the decoder output must be replaced with the corresponding source tokens. This *demasking* requires aligning the masks in the decoder output to the masks in the decoder input. Once this is done, recovering the original token identities for replacement is trivial. However, computing the mask alignment is not necessarily easy. We therefore explore two solutions to it: *indexing* and *bipartite matching*. Each of these solutions has its own benefits and problems.

Indexing The indexing approach (Crego et al., 2016) incorporates an index in each mask token:

EMAIL becomes EMAIL1, EMAIL2, and so on. Ideally, the decoder learns to output indexed mask tokens as a bijective permutation of the input mask tokens. The source tokens for each output mask are easily recovered in this scenario, but the downside is that there are now an unbounded number of masks which are all different to the decoder.

Bipartite matching Without indexes, we must produce our own alignment. We propose a general solution based on weighted bipartite matching. This approach takes as input a matrix of weights that assigns a score to each (source token, target token) pair. These weights can be obtained in different ways; for example, from the decoder attention weights, or from an external alignment model.

The task is to convert these weights into a set of hard alignments between the input and decoder output masks. We do this by formulating the problem as a bipartite graph problem (Algorithm 1. For each subset of masks with the same label, we use alignment scores as the edge weights, and execute the bipartite graph matching algorithm to find the best hard alignment scheme. These alignments can then be used to demask the output tokens.

Our approach guarantees an alignment for each target mask. If there are fewer target than source masks, an input token will be erroneously used multiple times.

Obtaining weights Obtaining weights to use with bipartite matching is not straightforward. We experiment with two approaches:

- *Averaged attention scores*. We average source attention scores across all decoder heads and layers in our model.
- *External aligner*. We run a version of fast-align (Dyer et al., 2013).

Both have problems. We use Transformers (Vaswani et al., 2017) in our experiments, but multi-head Transformer attention is not the same thing as alignment (Jain and Wallace, 2019). Fast-align is fast and easy to use at inference time, but it is a variant of IBM Model 2 (Brown et al., 1993) and the HMM model (Vogel et al., 1996). Therefore, its translation model cannot distinguish among mask permutations, and its impoverished distortion model is not well-suited to the task of recovering permutations of identical masks. However, we consider both approaches worth testing on this coarser alignment task, where we are only concerned with alignment of a handful of well-attested types, and not all the words in the sentence pair.

Algorithm 1: Bipartite Matching Demasking

```
Input: source sentence $\mathcal{S} = \{s_0, \dots, s_{I-1}\}$,
       target sentence $\mathcal{T} = \{t_0, \dots, t_{J-1}\}$,
       soft alignment matrix $\mathcal{A}$ of size $I \times J$
Output: demasked target sentence $\mathcal{T}'$
$\mathcal{T}' = \mathcal{T}$;
for each unique mask label $m$ in $\mathcal{T}$ do
    $\mathcal{C} = \emptyset$; // competing masks
    for $(s_i, t_j)$ in $\mathcal{S} \times \mathcal{T}$ do
        if $s_i, t_j$ are both masks and both
          belong to category $m$ then
            $\mathcal{C} = \mathcal{C} \cup \{(s_i, t_j)\}$;
        end
    end
    extract bipartite graph $\mathcal{G}$ corresponding to
      $\mathcal{C}$ using the weights from $\mathcal{A}$;
    conduct bipartite matching $\mathcal{M}$ on $\mathcal{G}$;
    for match $(s_i, t_j)$ in $\mathcal{M}$ do
        substitute $t_j$ in $\mathcal{T}'$ with the unmasked
          source token corresponding to $s_i$;
    end
end
return $\mathcal{T}'$;
```

4 Experiment Setup

4.1 Data

Our evaluation follows the WMT 2019 Robustness Task,[1] except that we use MTNT data (Michel and Neubig, 2018) *for evaluation only*. This includes MTNT/train, which we excluded from training in part because many of the masked items we would like to evaluate occur most frequently in this dataset. Table 2 contains information about all data sets.

For French–English training data, we use Europarl (Koehn, 2005, v7) and News Commentary (v10), and a portion of the UN Corpus. Due to its large size, we do not add all of the UN data, but add only lines that have a mask other than NUMBER, which includes about 1.1 million lines. This is crucial for the experiments since there is not enough masked data without this addition. We also include the WMT 2015 newstest test set for evaluation.

We also conduct limited experiments on Japanese–English. We follow Michel & Neubig in combining KFTT (Neubig, 2011), JESC (Pryzant

[1] www.statmt.org/wmt19/robustness.html

Dataset	French–English		Japanese–English	
	sents	words	sents	words
Europarl v.7	2.0m	50.2m	-	-
News commentary v.10	200k	4.4m	-	-
UN (complete)	12.8m	316.2m	-	-
→ UN (dict masks)	1.1m	33.8m	-	-
KFTT	-	-	440k	9.7m
JESC	-	-	3.2m	21m
TED Talks	-	-	241k	4.0m
newstest2014	3,003	69k	-	-
MTNT1.1/valid	886	34k	965	19k
newstest2015	1,500	25k	-	-
MTNT1.1/train	19k	660k	6,506	128k
MTNT1.1/test	1,022	16k	1,001	11k

Table 2: Pre-tokenization data sizes in sentence and English words for FR–EN and JA–EN training (top), validation (middle), and testing (bottom).

et al., 2017), and TED Talks (Cettolo et al., 2012) data.

4.2 Masks

We obtain our set of mask types from two sources: a set of regular expressions, and a dictionary extracted from the training data.

Regular expressions We built a set of regular expressions to identify the following mask types: NUMBER, EMOJI, EMAIL, and URL.

A difficulty with developing these regular expressions is their interaction with other steps in the pipeline. One first has to choose whether to apply masking before or after tokenization. A natural place is afterwards, but this requires that the tokenizer not split up the items we wish to mask, which in turn requires one to apply a set of regular expressions to exempt portions of the input segment.[2] As a result, we apply all masks to the raw data and modify tokenization and subword splitting code to not split up masks.

Dictionary We also want to test how well the system translates named entities. We identify these items by running the Stanford NER tagger on the English side of all the training data (including the complete UN corpus). We then construct a dictionary from all entries satisfying the following constraints, which simplify the masking and demasking process. Each entity:

- should be labeled as one of the following category: PERSON, LOCATION, ORGANIZATION, CITY, COUNTRY;
- must be found verbatim in the non-English side of the parallel sentence;[3] and
- must contain at least one word not among the most frequent 10k words in the training data.

Table 3 shows the statistics of pass-through items in MTNT dataset captured by our regular expression and named entity dictionary.

4.3 Synthetic Data

A problem apparent from Table 3 is that there simply aren't many instances for many of the mask types, which impedes investigation. MTNT/train has the most examples for many types, but for EMAIL, URL, COUNTRY, and even CITY, there are fewer than 1k, and often barely any at all.

To address this problem, we synthesize larger tests that allow us to see how often various types are translated correctly in the baseline system. For each mask, m, we identify all sentence pairs (s, t) in the training data for which one of the words was masked as m, ensuring the mask is in both the source and reference. Call this set D_m. Next, we build a set V_m of all tokens that get masked as m:

$$V_m = \{p \mid \text{mask(p)} = m\}$$

[2] The Moses tokenizer, which is applied with default settings in many scenarios, segments URLs into many pieces, due to a weak and buggy "protected patterns" file.

[3] This requirement limits our ability to identify Japanese entities, but it prevents errors from transliteration and/or alignment

Mask	French–English				Japanese–English		
	train	WMT15	MTNT$_{\text{test}}$	MTNT$_{\text{train}}$	train	MTNT$_{\text{test}}$	MTNT$_{\text{train}}$
Number	1,926,726	210	238	16,562	64,635	121	1,014
Emoji	5,434	1	5	131	2,057	11	352
Email	20,751	0	0	0	1	0	0
URL	38,655	0	0	26	175	0	5
City	186,902	39	16	824	13	0	0
Country	34,205	1	0	7	12	0	0
Location	409,109	41	24	1,598	155	0	2
Org.	369,297	73	46	1,896	507	0	40
Person	845,116	131	60	3,395	179	0	3

Table 3: Entity counts across all data. For training data, the counts are "true" counts, that is, they are only counted for tokens that appeared on both the source and target sides of the data. For test sets, the counts are produced by matching only against the source. For most entity types, data is quite sparse.

We then produce a new test set by repeating the following procedure 5,000 times:

1. Sample a sentence pair $d \in D_m$;
2. Twenty separate times, do
   (a) Sample one of the positions with mask m in d (there may be only one);
   (b) Sample a term $s \in V_m$;
   (c) Create a new sentence pair by inserting s into d.

This yields synthetic datasets of 100k sentences. Table 7 contains examples.

4.4 Models

Our baseline NMT system is a 4-layer transformer trained with Sockeye (Hieber et al., 2017). We use the following settings for training both French–English and Japanese–English models: eight attention heads, model size of 512, feed-forward layer size 2048, three-way tied embeddings, layer normalization applied before attention, dropout and a residual connection added afterwards, a batch size of 4096 words, and the learning rate initialized to 0.0002. We compute checkpoints every 5000 updates, and train until validation likelihood does not increase for ten consecutive checkpoints.

For preprocessing, we first apply the Moses scripts that normalize punctuation, remove non-printing characters, and tokenize.[4] We learn a subword model using byte-pair encoding (Sennrich et al., 2016) with 32k merge operations. No recasing is applied to either source- or target-language text.

For alignment-based demasking, we trained two fast-align models, one in each language direction, using default parameters. We then combine them with the `grow-diag-final-and` heuristic.

Source Factors We also experiment with source factors (Sennrich and Haddow, 2016) applied to the baseline (unmasked) system. Source factors are separate embeddings that are learned from annotations applied to the input tokens. For each of the types Number, Email, and URL, instead of masking, we added a distinct binary source factor. We also experimented with two ways of combining factors: *concatenation* and *summing*. Concatenation was described in Sennrich et al.; we learn an embedding of size 4 for each factor, and concatenate with the subword embeddings. For summing, we instead embed each factor to size 512, and sum together all factors for each input token.

5 Results

We compute BLEU on detokenized, cased outputs using the standardized BLEU scoring script, sacreBLEU (Post, 2018).[5] The results on all test sets can be found in Table 4. We provide the same-data baseline score from Michel and Neubig (2018) as an anchor point for evaluating the models.

In no masking situation is there any improvement in BLEU score over the baseline system. In fact, adding masks seems to uniformly cost the models in

[4] With the options `-no-escape` and using a version of the Moses `basic-protected-patterns` file modified to protect masks.

[5] Shared portion of signature: `BLEU +case.mixed +numrefs.1 +smooth.exp +tok.13a +version.1.2.20`.

System	French–English			JA–EN	
	WMT15	$\text{MTNT}_{\text{test}}$	$\text{MTNT}_{\text{train}}$	$\text{MTNT}_{\text{test}}$	$\text{MTNT}_{\text{train}}$
Michel & Neubig (Base)	-	23.2	-	6.6	-
baseline	32.0	28.1	28.7	8.2	6.5
indexed masking	31.8	27.4	27.0	8.0	6.6
masking (fast-align)	31.9	27.9	*	8.1	*
masking (attention)	31.9	28.0	27.5	8.1	5.4
source factors (concat)	32.0	28.1	28.3	8.2	6.0
source factors (summed)	32.4	28.4	29.1	8.2	6.7

Table 4: BLEU scores on test sets. The score take from Michel & Neubig is the system *not* trained on MTNT/train, since we did not train on that in this paper, instead reserving it for analysis.

type	WMT	MTNT /test	MTNT /train	synth
Number	91.1	95.2	94.8	-
Emoji	*0*	*0*	5.2	-
Email	-	-	-	96.9
URL	-	-	*91.7*	91.3
City	*100*	*92.3*	95.1	98.4
Country	*100*	-	*50.0*	90.2
Location	*100*	*100*	87.9	-
Org.	98.4	*100*	93.6	-
Person	99.2	100	94.9	-

Table 5: FR–EN baseline recall scores (against the reference) for each data type when decoding with the baseline system. Hyphens (-) indicate no data being available, and *italics* indicate counts for which there were fewer than 50 instances (Table 3). The synthetic dataset is discussed in Section 6.3.

terms of BLEU score, from small drops of a tenth of a point or so (for WMT15 and Japanese), to large drops of about half a BLEU point on FR-EN MTNT. We do, however, see BLEU score increases of about a third of a point when using summed source factors.[6]

BLEU is important, but is too coarse of a metric to draw conclusions from in this situation that deals with relatively rare phenomena. We turn now to a more fine-grained analysis.

6 Baseline Analysis

We begin with an analysis of the performance of the baseline system on all the mask types in our study. Table 5 reports, for each type, the percentage of time that the baseline system correctly translated tokens that were in both the source and reference.

6.1 Number

Numbers are by far the most frequent category type, and additionally for many scenarios numbers are considered to be one of the data types that are important to correctly translate. How well are numbers translated?

On WMT15, there are 210 instances of numbers that are matched by our regular expression and exist in both the French input and the English reference. On these numbers, the baseline system achieves an accuracy of 91.2%, leaving only 18 instances of missed masks. Of these, the vast majority are fine: 12 are found in written form in the system output (e.g., *twelve* instead of 12), and four are localization effects of time (e.g., *14:30 → 2:30 PM*). Accounting for these, the accuracy is 99.0%.

Turning to MTNT/test, we find an accuracy of 92.2% on 219 masked instances, with 17 of them translated incorrectly. Of these, 11 are fine (written substitutions), and many are the result of the decoder entering a “language modeling mode”, where it generates output that has little to do with the input (Koehn and Knowles, 2017). A few are actually wrongly translated: *15 jours* gets translated as *fortnight*, and *1h de sommeil* (“one hour of sleep”) is mistranslated.

Finally, we look at MTNT/train, where there are many more masks, especially numbers. MTNT/train is an unusual dataset. There are many input segments with hundreds or over a thousand words, often containing multiple sentences, due to the way the data was collected (Michel and Neubig, 2018, penultimate paragraph of §3.4) There is also a lot of repetition: some input sentences

[6] Recall that these are applied only to numbers, email addresses, and URLs, and that these terms are not masked, but instead have the standard tokenization and subword-splitting regime applied to them.

type	# of digits 1	2	3	4	*
correct	203	9	0	1	2
missing	6	8	1	6	0
wrong	1	5	0	0	2
total	210	23	1	87	4

Table 6: Counts of error types made by the baseline system on FR–EN MTNT/train on 1-, 2-, 3-, and 4-digit integers, and other numbers (*), looking only at system outputs with 50 or fewer words. *missing* and *wrong* denote errors where the number is either dropped or mistranslated by the baseline system. *correct* sentences were fine but not identical (e.g., "1,000" and "1000" or "1" and "one").

are repeated three or four times, leading to skewed statistics. It is also quite informal, and since we had no such data in our training data, we often observed the NMT model again entering "language model mode". The accuracy is 95.7% on 10,040 instances with system outputs with 50 or fewer words.

We analyze the 325 instances where our method reports an error (Table 6).[7] The error counts are produced by counting all instances where a number matching our pattern is found in both the French source and English reference, but not in our system output. We break down the analysis by number type: integers with one to four digits, and all others.

It is clear that the analysis from above holds: the majority of items marked incorrect by automatic matching are actually fine (65%). The six missing 4-digit numbers seem to be a quirk of the data: six of the source sentences have X *Edition* at the start of the input and reference (for some year X), with no punctuation, and it gets dropped. The handful of other errors are similar to those described above. If we remove the bad lines, and count as correct the sentences we identify, the new recall for numbers on MTNT/train is 98.8%.

The baseline JA-EN system does not perform nearly as well as the FR-EN system. The accuracy for numbers is only 49.3% on MTNT/test and 61.2% on MTNT/train. However, we see the same pattern of mismatches that are not errors (e.g., numbers spelled out or formatted slightly differently). Accounting for these, the recall on MTNT/test jumps to 67.1%. This is still much lower than we see for French, but not unexpected given the drastic difference in BLEU score.

The bottom line on these test sets is that numbers appear to be correctly passed-through or translated the majority of the time in the high resource setting. They are also often correctly translated in context-sensitive ways. However, they are not perfect.

6.2 EMOJI

We use the term *emoji* broadly to indicate special characters that are outside the phonetic alphabet. Emoji are a unique type of data, because they are typically single Unicode codepoints. If these codepoints were not in the training data, they will be untranslatable. This is precisely what happens in WMT15, where the single instance

> *L'introduction mi-septembre par Apple^{TM} d'écrans plus grand pour...*

is mistranslated. Emoji are therefore a unique candidate for masking.

6.3 EMAIL, URL, CITY, and COUNTRY

These four categories have almost no data in the test sets, so we instead analyze the synthetic data (§4.1). The synthetic data provides us with 5,000 sentence contexts with 20 different instances, totalling nearly 100k samples (Table 7). We translate each of these sentences with the baseline system, and check whether the entity type is in the system output. The results can be found above in the last column of Table 5.

We focus here on EMAIL and URL. Note that these are types which should almost always be passed through, and not translated. Yet the baseline system mistranslates 3.1% of email addresses and 8.7% of URLs. The reason likely has to do with the MT preprocessing pipeline: both tokenization and subword processing mangle these types into long sequences of tokens. On average, URLs are transformed into 14.1 subword tokens (the longest is 125 tokens), versus 3.9 subword tokens for the average vocabulary item.

Looking at the outputs, we see that URLs are usually translated nearly perfectly, except for a small mistranslated or dropped piece (Table 8). But for these types, a single character mis-translation renders the entire item useless.

6.4 LOCATION, ORG., and PERSON

These three categories are a bit unusual, since we are restricting our attention to instances that have the same surface form in both French and English (instead of using a translation dictionary). All of

[7] This is after throwing out 103 instances where the input was multiline or the NMT output was garbage, perhaps due to out-of-domain effects.

Présidence de l'Union européenne : http://europa-eu-un.org
Présidence de l'Union européenne : http://www.fao.org/figis/servlet/static?xml=CCRF_prog.xml&dom=org&xp_nav=2,3
Présidence de l'Union européenne : www.all4syria.org
Présidence de l'Union européenne : http://www.njcl.fi/1_2006/commentary1.pdf

Prière de prendre contact avec le Groupe du Journal, à l'adresse journal@un.org.
Prière de prendre contact avec le Groupe du Journal, à l'adresse frank.X@univie.ac.at.
Prière de prendre contact avec le Groupe du Journal, à l'adresse jferex@eclac.cl.
Prière de prendre contact avec le Groupe du Journal, à l'adresse chungrx@un.org.

Table 7: Substitutions for URL (top) and EMAIL (bottom). The original is in bold. Personal email addresses have been slightly modified.

sys:	http://www.cbs.nl/NR/rdonlyres/D1716A60-0D13-4281-BED6-3607514888AD/
ref:	http://www.cbs.nl/NR/rdonlyres/D1716A60-0D13-4281-BED6-360751488AD/
sys:	www.fao.org/forestry/fo/fra/index.jsp
ref:	www.fao.org/forestry/fo/fra/index.jp
sys:	qualityws.ht
ref:	qualitativeyws.ht
sys:	http://www.tebtebba.org/tebtebba_files/ipr/racism.htm
ref:	http://www.tebba.org/tebtebba_files/ipr/racism.htm

Table 8: Mistranslated URLs.

them display similar patterns: extremely high accuracies in all three test sets (WMT15, MTNT/test, and MTNT/train). We took the most prevalent category, PERSON, and manually examined the error cases. Of the 131 tokens tagged as PERSON in MTNT/test, seven did not appear in the reference, leaving 123 instances, on which the baseline system achieved 99.2% accuracy, missing only one. The single missed instance translated *Jean-Pierre Bernajuzan* as *Mr Bernajuzan*.

No mistakes were made on MTNT/test. MTNT/train is more difficult to analyze, but many of the missing instances were caused by multiline inputs where the NMT system stopped translating after the first sentence of the input.

In summary, for these categories, the baseline system does very well. But again, it's not perfect.

type	**indexed**		**unindexed**		
	1	2+	1	2+	3+
NUMBER	98.5	95.9	98.5	97.7	97.7
EMOJI	91.4	74.0	98.8	92.0	100
EMAIL	-	-	-	-	-
URL	100	-	100	-	-
CITY	98.9	97.6	97.6	97.1	96.4
COUNTRY	100	100	100	100.0	-
LOCATION	98.6	92.0	99.0	93.6	90.7
ORG.	98.9	90.7	98.2	93.8	91.7
PERSON	98.3	82.1	98.5	97.3	96.7

Table 9: FR–EN recall scores (against the reference) for masking on MTNT/train, broken down between indexing and (attention-based) not-indexing, and between sentences that have only a single (1) or multiple (2+) instances of a mask.

6.5 Source Factors

We applied source factors to types NUMBER, EMAIL, and URL in the baseline system. From Table 4, this seems to have had no effect on BLEU scores when using the embedding concatenation described in (Sennrich and Haddow, 2016), except for a minor drop on MTNT/train. When summing the factors, however, we see a small improvement in BLEU score on all three test sets. However, there was no improvement in entity-based recall scores over the baseline analysis in Table 5.

7 Masking

Masking has the potential to achieve 100% accuracy on masked entities. However, its success depends on a number of pieces: (1) the masks need to be translated correctly (i.e., one-for-one with the input masks), and (2) for unindexed masking, they need to be correctly aligned.

Table 9 looks into (1). It displays masks recall scores on the MTNT/train test set, broken down between indexed and unindexed masking, and between sentences with exactly one instance of each mask, or more than one (2+). For unindexed masking, we also display recall for masks appearing 3+ times in a single sentence. We see that masks are not perfectly translated, but that unindexed masking does a slightly better job of it. And the numbers are somewhat better than those of the baseline system in Table 5, though for some labels they are not that different. Performance degrades with more masks of the same type, in all instances except EMOJI (where there are only 18 3+ instances).

		reference		reference	
		mono	not	mono	not
system	mono	61	18	67	20
	not	7	2	1	0

Table 10: Demasked permutations for the attention-based (left) and alignment-based (right) approaches. Mono/not denotes whether the text of the decoder output (rows) and reference (columns) was monotonic with respect to the input.

Demasking Section 3.1 described two approaches to aligning masks: decoder attention and post-alignment via fast-align. This use case proves similarly difficult to analyze for a number of reasons. On WMT15 (where we expect the neat text to present the simplest case), there are 88 instances where a single mask type appears more than once in a sentence. We break down the analysis into whether or not the permutations of the text in the (a) system output and (b) reference were monotonic (with respect to the input text). (Note that in the case of non-monotonic permutations, we are not guaranteed that the system and reference line up.) The results are in Table 10, and are largely inconclusive. There is not a lot of data to determine whether permutations are correctly restored, and there does not appear to be much difference between the two approaches.

8 Conclusions

We began this paper wondering whether "translation guarantees" for certain word types were necessary, and whether masking was an appropriate tool for guaranteeing them. The answer is not as clear-cut as we would have liked. Masking (or placeholding) is sometimes viewed as a way of ensuring or increasing the chances that a particular entity type is correctly translated. Our experiments on different test sets with a modern Transformer architecture on French–English and (to a lesser extent) Japanese–English show that this is often not the case. Masked systems do not reliably translate masks, which is likely why Crego et al. mention the use of constraints to ensure masks are output. And in any case, the baseline system does a decent job of translating many of these types already. The recall numbers between the baseline and masked systems (Tables 5 and 9) all range in the mid-90s across multiple test sets.

Another issue is that the set of items that should be masked cannot be perfectly predicted. As we saw with types like NUMBER, many numbers should not in fact be passed through, but require translation, in ways that are often mediated by context. Using masks for such types is akin to a vote of "no confidence" in the decoder, which seems not to be justified. This also seems to be the case for other entity-based types, which are handled well by the baseline system.

However, we have seen that unindexed masking can do a good job of passing items through, compared to Crego et al. (2016)'s indexed system. In situations where it is better to drop the identified term than to mistranslate it, unindexed masking may be preferable. This includes terms like emojis and extended character sets, and email addresses and URLs. The former are important to mask because otherwise the characters will be outside the decoder character set; one could alternately augment the training data with all emoji types, but this could be difficult and error-prone, especially as new characters are introduced all the time. Email addresses and URLs cause complications with tokenization, can get broken up into many subword pieces, and can also be hard to reliably detokenize. It makes sense to translate these items as a single entity, making masking the clear option for this.

There are many avenues we have not explored in this paper. For example, adding a source factor to masked tokens might help increase the reliability of mask translation. An even better approach may be to use special loss functions to further encourage the decoder to get marked tokens right. One could also use constrained decoding (Hokamp and Liu, 2017; Post and Vilar, 2018) to ensure that desired items (or masks) are placed in the output.

References

Anderson, Peter, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark, September. Association for Computational Linguistics.

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

Cettolo, Mauro, Christian Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. In *Conference of European Association for Machine Translation*, pages 261–268.

Chatterjee, Rajen, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168, Copenhagen, Denmark, September. Association for Computational Linguistics.

Crego, Josep Maria, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems. *CoRR*, abs/1610.05540.

Dyer, Chris, 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, June. Association for Computational Linguistics.

Gotti, Fabrizio, Philippe Langlais, and Atefeh Farzindar. 2014. Hashtag occurrences, layout and translation: A corpus-driven analysis of tweets published by the canadian government. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 2254–2261.

Grundkiewicz, Roman and Kenneth Heafield. 2018. Neural machine translation techniques for named entity transliteration. In *Proceedings of the Seventh Named Entities Workshop, NEWS@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 89–94.

Hasler, Eva, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana, June. Association for Computational Linguistics.

Hieber, Felix, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *CoRR*, abs/1712.05690.

Hokamp, Chris and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada, July. Association for Computational Linguistics.

Jain, Sarthak and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota, June. Association for Computational Linguistics.

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

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Li, Zhongwei, Xuancong Wang, AiTi Aw, Eng Siong Chng, and Haizhou Li. 2018. Named-entity tagging and domain adaptation for better customized translation. In *Proceedings of the Seventh Named Entities Workshop, NEWS@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 41–46.

Long, Zi, Takehito Utsuro, Tomoharu Mitsuhashi, and Mikio Yamamoto. 2016. Translation of patent sentences with a large vocabulary of technical terms using neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation, WAT@COLING 2016, Osaka, Japan, December 2016*, pages 47–57.

Luong, Thang, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China, July. Association for Computational Linguistics.

Michel, Paul 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. Association for Computational Linguistics.

Neubig, Graham. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Post, Matt and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana, June. Association for Computational Linguistics.

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

Pryzant, Reid, Yongjoo Chung, Dan Jurafsky, and Denny Britz. 2017. JESC: japanese-english subtitle corpus. *CoRR*, abs/1710.10639.

Radford, Will, Ben Hachey, Bo Han, and Andy Chisholm. 2016. : telephone: : person: : sailboat: : whale: : okhand: ; or ”call me ishmael” - how do you translate emoji? In *Proceedings of the Australasian Language Technology Association Workshop 2016, Melbourne, Australia, December 5 - 7, 2016*, pages 150–154.

Sennrich, Rico and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany, August. Association for Computational Linguistics.

Sennrich, Rico, 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, August. Association for Computational Linguistics.

Ugawa, Arata, Akihiro Tamura, Takashi Ninomiya, Hiroya Takamura, and Manabu Okumura. 2018. Neural machine translation incorporating named entity. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3240–3250.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.